ML101

機器學習教材

PYTHON在AIOT的系統實作：機器學習

課程內容

◆課程大綱

- Python開發與環境
- 機器學習基礎實務

◆ 課程內容：

7/09 (星期二) 北社工18樓第九教室		
時間	講題	講師
09:00 ~ 12:00	Python簡介與開發環境架設、資料清理 說明與操作演練	王宏仁、梁瑋泓
12:00 ~ 13:30	Break	
13:30 ~ 16:30	資料清理 實作與討論	李婕、鍾亞凌、吳則諭
7/16 (星期二) 北社工18樓第九教室		
時間	講題	講師
09:00 ~ 12:00	特徵工程、基礎模型建立 說明與操作演練	王宏仁、梁瑋泓
12:00 ~ 13:30	Break	
13:30 ~ 16:30	特徵工程、基礎模型建立 實作與討論	李婕、鍾亞凌、吳則諭
7/23 (星期二) 北社工18樓第十教室		
時間	講題	講師
09:00 ~ 12:00	模型參數調整、真實資料實作演練 說明與操作演練	王宏仁、梁瑋泓
12:00 ~ 13:30	Break	
13:30 ~ 16:30	模型參數調整、真實資料實作演練 實作與討論	李婕、鍾亞凌、吳則諭

PYTHON在AIOT的系統實作：機器學習

課程目標

學會使用一種工具，並且了解背後的原理是什麼

- 建立可以隨時進行 機器學習專案 的開發環境。
- 了解「程式設計思維」的步驟並學習解決問題的能力
- 進行 Python 程式開發和程式語言的基礎。
- 了解資料工程師與資料科學家的職責和技能區別。
- 機械學習的原理和整體基礎概念。
- 資料清理的基礎概念和實作。

PYTHON在AIOT的系統實作：機器學習

程式設計思維

學習解決問題的能力

1. 尋找並發現問題。
2. 釐清並定義問題。
3. 分解問題。
4. 尋找解決方法和資源。
5. 驗證問題和解決方式。

PYTHON在AIOT的系統實作：機器學習

PYTHON 程式設計

大綱

- Python 語言簡介
- 開發環境的安裝和簡介
- Git和GitHub版本控制系統簡介
- 虛擬環境的創建
- 第一個 Python 程式
- Markdown簡介
- Python 基礎教學

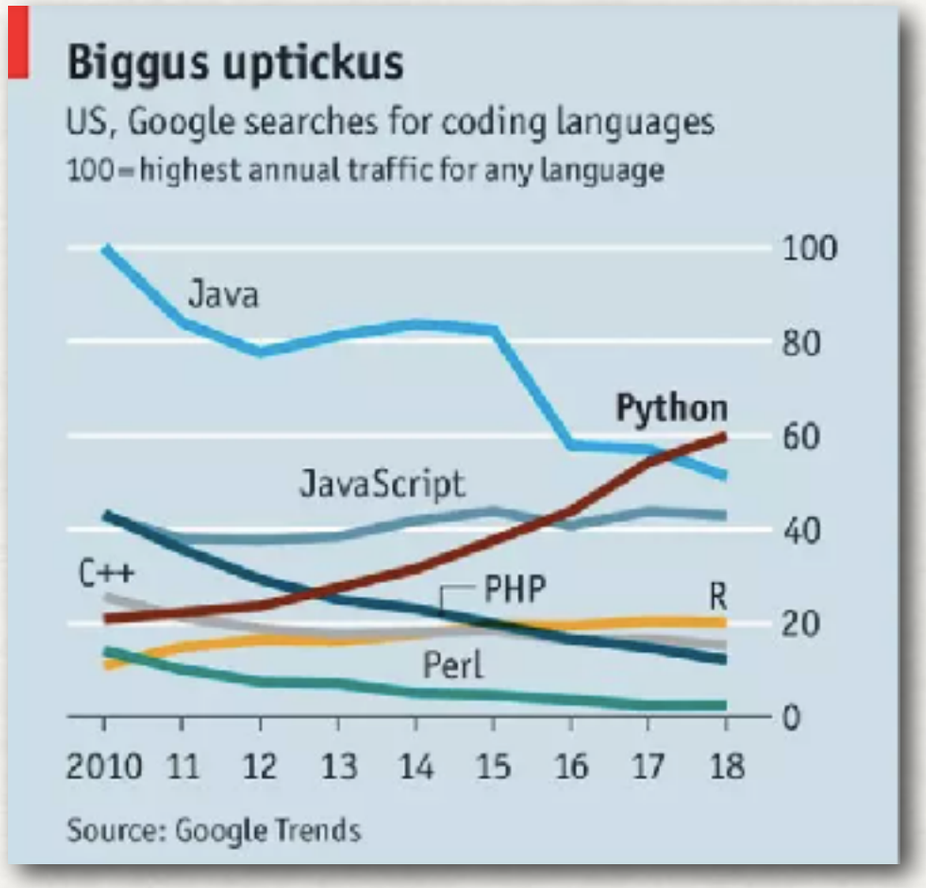
PYTHON

經濟學人：「 *Python* 是編程語言中最*skr*的。」

查詢率

- 語法簡單，易於學習。
- 開源平台，便於共享; 以及其擁有但龐大第三方軟體包 。
- Python 已成為人工智慧研究人員的首選語言，他們也為此製作了大量的軟體包 。

所以 PYTHON 已經成為人工智慧領域中擁有豐富套件並且可以快速開發的程式語言。



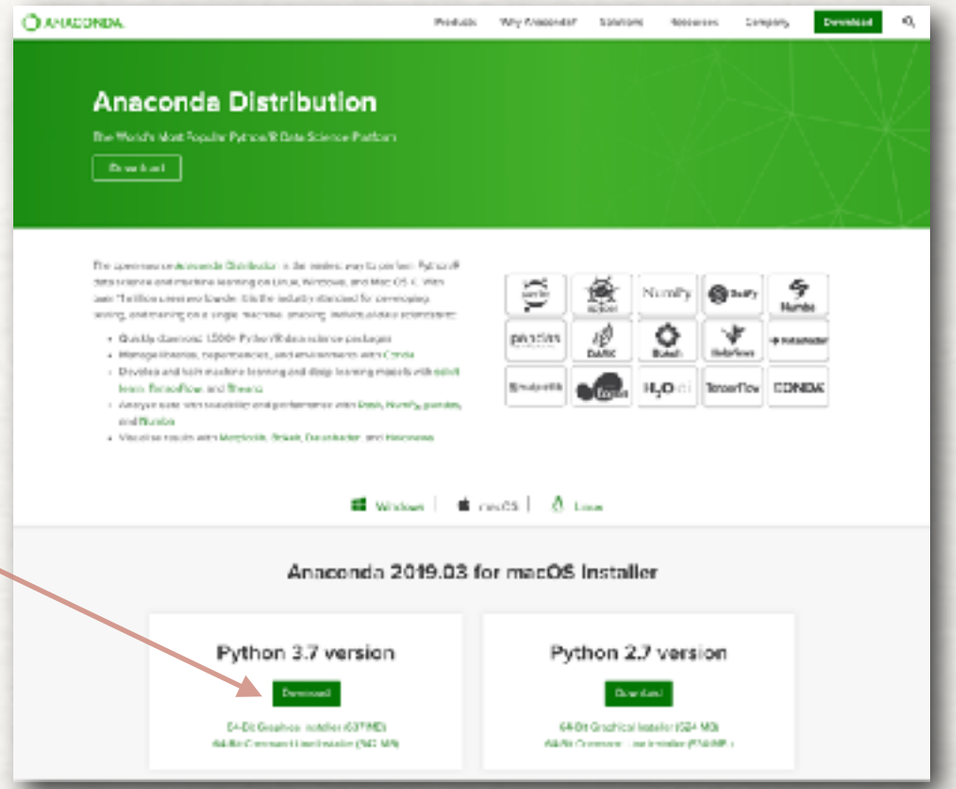
開發環境

Anaconda

- All-in-one 的 Python 開發環境
- 便於安裝各種科學、數學、工程、數據分析模組。
- 開源並且免費。
- 跨平台支持：Linux、Windows、Mac
- 內建 Spyder 和 Jupyter Notebook
- 安裝流程簡單
- 官網：https://www.anaconda.com/

點擊右上角下載連結

選擇3.7版本下載



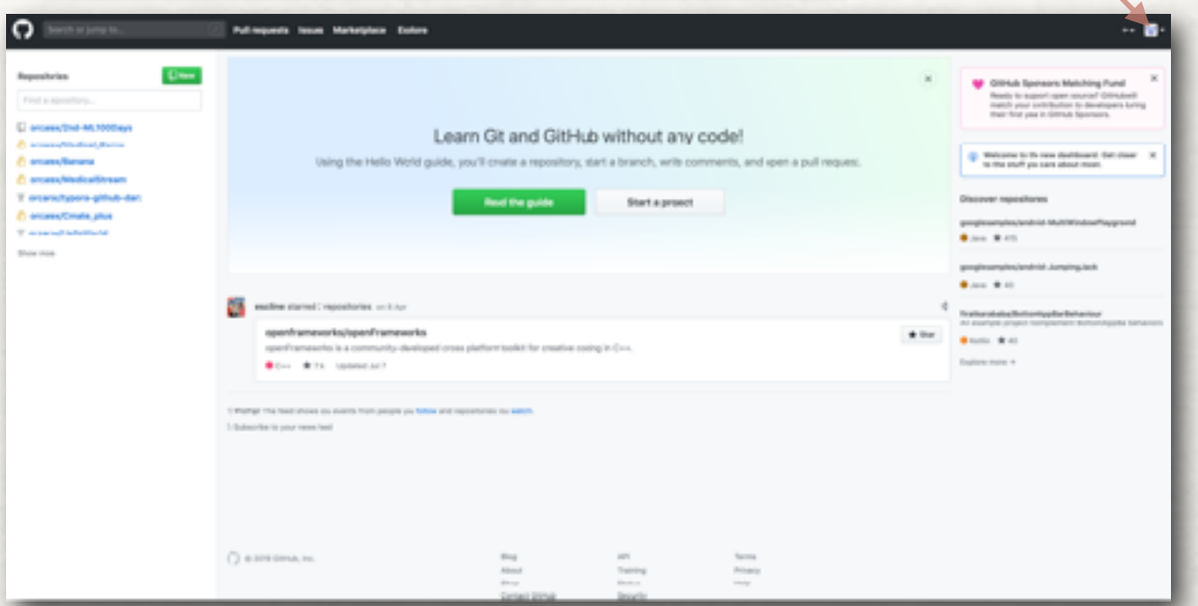
版本控制

Git & GitHub

- Git 是一種分散式版本控制系統，可以讓我們更方便的管理我們的程式碼
- GitHub 是一個可以存放 Git 程式碼專案的平台，我們可以從這裡接觸到最新的開源程式碼資訊
- Git 官網：https://git-scm.com/
  GitHub 官網：https://github.com/
- 網路上有相當多優秀的文件（連猴子都能懂的Git入門指南、寫給大家的 Git 教學、初心者 Git 上手攻略）

下載並安裝GIT

註冊 註冊並登入帳號



版本控制

Git & GitHub 簡單指令

複製專案

- Git clone https://github.com/orcarex/AIOT_class.git
- Cd AIOT_class

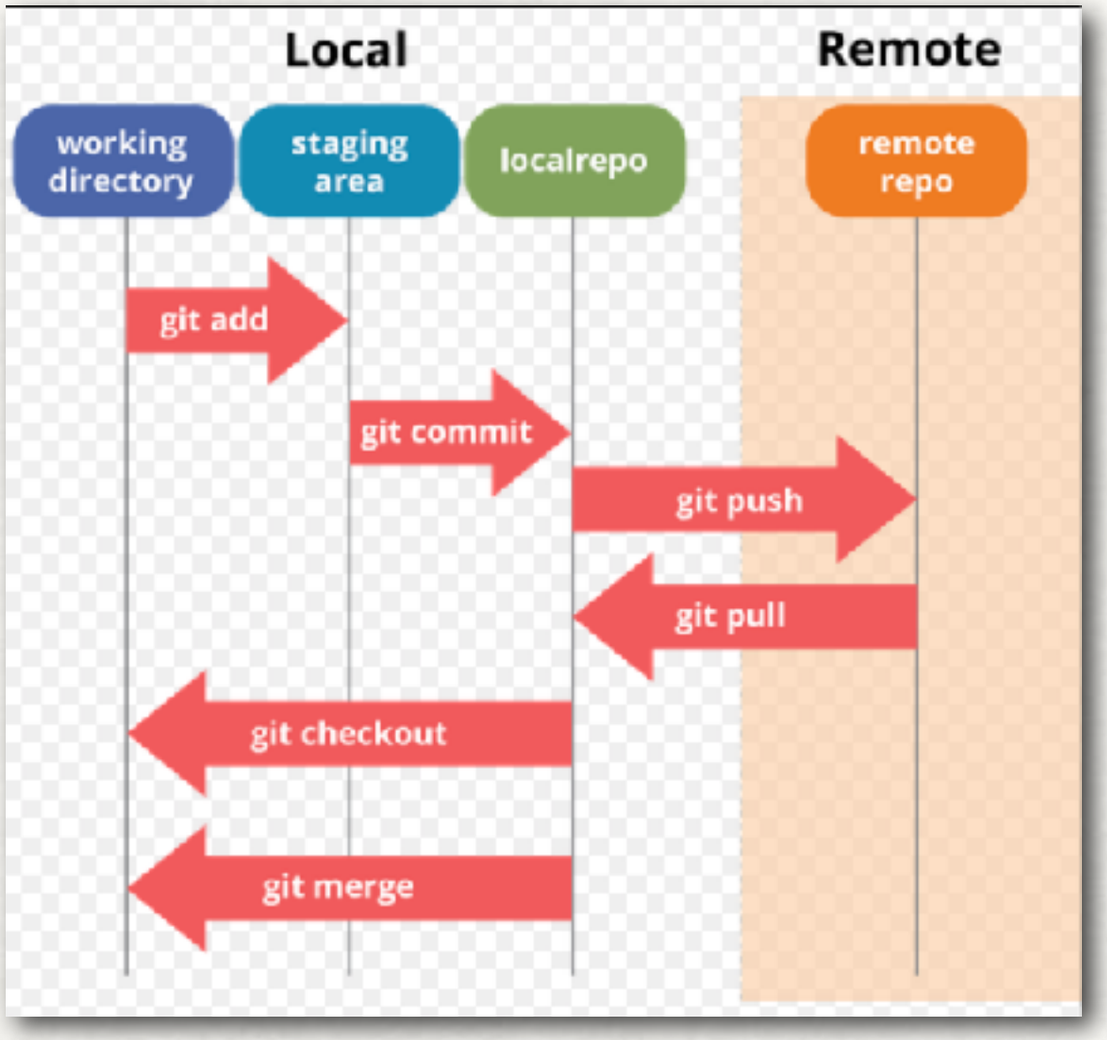
將所有變更檔案放入暫存區
使用 git add . 或是 git add *

常見指令

```
# 初始化專案
$ git init
# 查看狀態
$ git status
# 檢查差異
$ git diff
# 將變更檔案放入暫存區
$ git add index.py
# 使用 commit -m 提交變更
$ git -a -m 'init commit'
# 查看歷史
$ git log
# 放棄已經 commit 的檔案重回暫存區
$ git reset HEAD index.py
# 放棄檔案變更
$ git checkout index.py
```

版本控制流程

Git & GitHub 工作區域和指令



Git工作區域示意圖

常用工作指令

```
# 查看狀態
$ git status
# 將所有變更檔案放入暫存區
$ git add .
# 使用 commit -m 提交變更
$ git -a -m ‘some messages'
# 查看歷史
$ git log
# 將repo推送至遠端更新
$ git push
# 將遠端repo更新至local端
$ git pull (同等於git fetch + git merge)
# 將某歷史版本從localrepo更新至工作區
$ git checkout
```

虛擬環境

建立一個 *Python* 的專案

建立專案資料夾

```
$ cd ~/Desktop
```

移至目錄

```
$ mkdir python_examples
```

創建資料夾

```
$ cd python_examples
```

虛擬環境

```
$ conda create -n
python_examples_venv python

$ source activate
python_examples_venv
```

常用指令

```
# 顯示目前虛擬環境列表

$ conda info -e

# 創建虛擬環境

$ conda create -n 套件名稱 python=3.6

# 進入虛擬環境（若是 Windows cmder 環境不用
加 source），成功後提示字元變成：（套件名稱）
$

$ source activate 套件名稱

# 離開虛擬環境（若是 Windows cmder 環境不用
加 source）

$ source deactivate
```

安裝anaconda相關套件
可進入環境後再安裝

環境名稱

$ conda create --name tensorflow anaconda

進入環境後
使用 conda install (官方建議)
或是 pip install 安裝欠缺套件

第一個 PYTHON 程式

隨機印出不同語言

步驟

1. 引入 random 套件
2. 定義 language 變數，並將儲存程式語言字串的串列（list）資料結構給定給變數
3. 產生 1 到 2 之間的隨機整數
4. 列印出最喜歡的程式語言字串（含根據隨機產生的 index 選取到的串列值）

程式碼

```
import random

language = ['Python', 'JavaScript', 'Java']

rnd = random.randint(1, 2)

print('My favorite Language is' + language[rnd])
```

常用套件

```
# 隨機套件
$ import random
# 資料集
$ from sklearn import datasets
# 畫圖套件
$ import matplotlib.pyplot as plt
# 數據分析套件
$ import pandas as pd
$ import numpy as np
# 決策樹分析
$ from sklearn.tree import DecisionTreeClassifier
```

第一個 PYTHON 程式

利用*Requests*套件爬蟲

步驟

1. 引入 requests 套件
2. 爬取政府公開 Wifi 熱點資料，將取得資料回應給定給 response 變數
3. 將資料轉換成以 { "key": "value "} 形式的 json 格式
4. 印出取得的資料

程式碼

```
import requests

url = 'http://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=b087af42-2c54-4dbf-86c8-427d788779e5'

response = requests.get(url)

data = response.json()

print(data)
```

提升生產力的輕量標記語言

利用Markdown做筆記

簡介

- Markdown是一種輕量級的「標記語言」，它的優點很多，目前也被越來越多的寫作愛好者，撰稿者廣泛使用。看到這裡請不要被「標記」、「語言」所迷惑，Markdown的語法十分簡單。常用的標記符號也不超過十個，這種相對於更為複雜的HTML標記語言來說，Markdown可謂是十分輕量的，學習成本也不需要太多，且一旦熟悉這種語法規則，會有一勞永逸的效果。

優點

- 專注你的文字內容而不是排版樣式，安心寫作。
- 輕鬆的導出HTML、PDF 和本身的.md 文件。
- 純文本內容，兼容所有的文本編輯器與字處理軟件。
- 隨時修改你的文章版本，不必像字處理軟件生成若干文件版本導致混亂。
- 可讀、直觀、學習成本低。

簡要規則

- 標題 # ## ###
- 列表 *
- 引用 >
- 圖片 、 連結 []()
- 粗體 ** ** 、 斜體 * *
- 表格
- 代碼框 ```
- 分隔線 ***

```
| Tables        | Are          | Cool  |
| ------------- |:-------------:| -----:|
| col 3 is      | right-aligned | $1600 |
| col 2 is      | centered      |   $12 |
| zebra stripes | are neat      |    $1 |
```

這種語法生成的表格如下：

Tables	Are	Cool
col 3 is	right-aligned	$1600
col 2 is	centered	$12
zebra stripes	are neat	$1

PYTHON基礎教學

The Zen of Python

哈囉世界

- import this
- 禪(Zen)：寫程式的coding rules
  試著用最優雅的方式寫出程式

基礎語法

- 變數
- 註解
- User Input
- 數學運算子
- 流程控制
- 比較運算子
- 布林運算

PYTHON之禪

```
In [44]: import this

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

翻譯節錄

- 可讀性很重要，除了電腦，你的同事 (還有未來的你)也要讀你的程式碼!
- 再小的錯誤與例外，也要好好處理，切勿養虎為患。
- 一個難以解釋清楚的實作? 那真是太糟啦! 一個可以輕易解釋的實作? 嘿! 這說不定是個好實作呢!

PYTHON基礎教學

Data Structure

基礎物件

bool, int, float and string

- 在python的世界中，所有東西都是物件。
- 可使用type(物件)確認該物件的型別。

不可變物件

- int
- string
- float
- Tuple

可變物件

- List
- Dict
- Set

功能物件

List, tuple, dict and set

- list 串列，用於儲存一系列序列資料（可以儲存不同資料型別），類似於其他程式語言的陣列但更為強大。
- tuple 元組，類似於 list 串列，但不同的是 tuple 給定元素後不能改變。
- dict 字典，和串列很像但不在乎元素順序，而且不會使用 0,1,...等的序列 index 來選擇項目，反之我們必須宣告唯一的 key 鍵值來對應想儲存的 value 值。
- set 集合，集合就像是被移除 value 值的字典，只有保留 key 鍵值，也就是說 set 的內容元素都必須是獨一無二的。

PYTHON基礎教學

Practice

- 物件型別判定與型別變換
- 數學運算
- 字串操作
- 使用者輸入
- 比較運算
- 流程控制
- 猜數字
- 猜拳遊戲

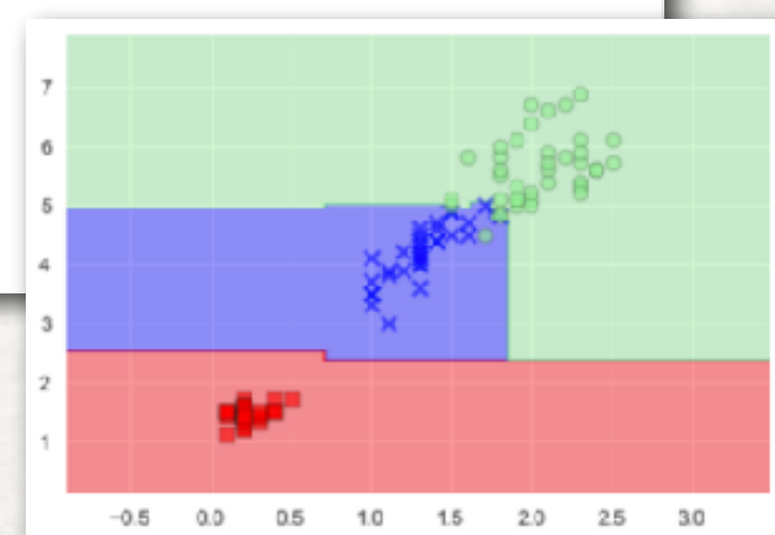
猜數字

```
import random
ans =  random.randint(0,10) #取從1到10的亂數
times = 0
print('請猜一個0-10的數字')
play = True

while(play):
    try:
        guess = int(input('來隨便猜一個數字吧： ')) #input出來預設為str，轉型為int
    except:
        print('程式出現非預期的錯誤，遊戲結束，答案是'+str(ans))
        #str()是將數字轉文字，'+'串接的必須是同樣的型態 -
        play = False
    else:
        times = times+1 #每猜一次就要+1
        if (guess > ans):
            print('喔你猜得太大囉')
        elif (guess < ans):
            print('喔你猜得太小囉')
        else:
            print('恭喜你猜對了！')
            play = False
    finally:
        print('你猜了'+str(times)+'次')
```

```
請猜一個0-10的數字
來隨便猜一個數字吧： 6
喔你猜得太大囉
你猜了1次
來隨便猜一個數字吧： 4
喔你猜得太大囉
你猜了2次
來隨便猜一個數字吧： 2
喔你猜得太大囉
你猜了3次
來隨便猜一個數字吧： 1
恭喜你猜對了！
你猜了4次
```

- plus：迴圈
- plus：lambda, map, filter
- plus：決策樹, 隨機森林



PYTHON在AIOT的系統實作：機器學習

機器學習簡介

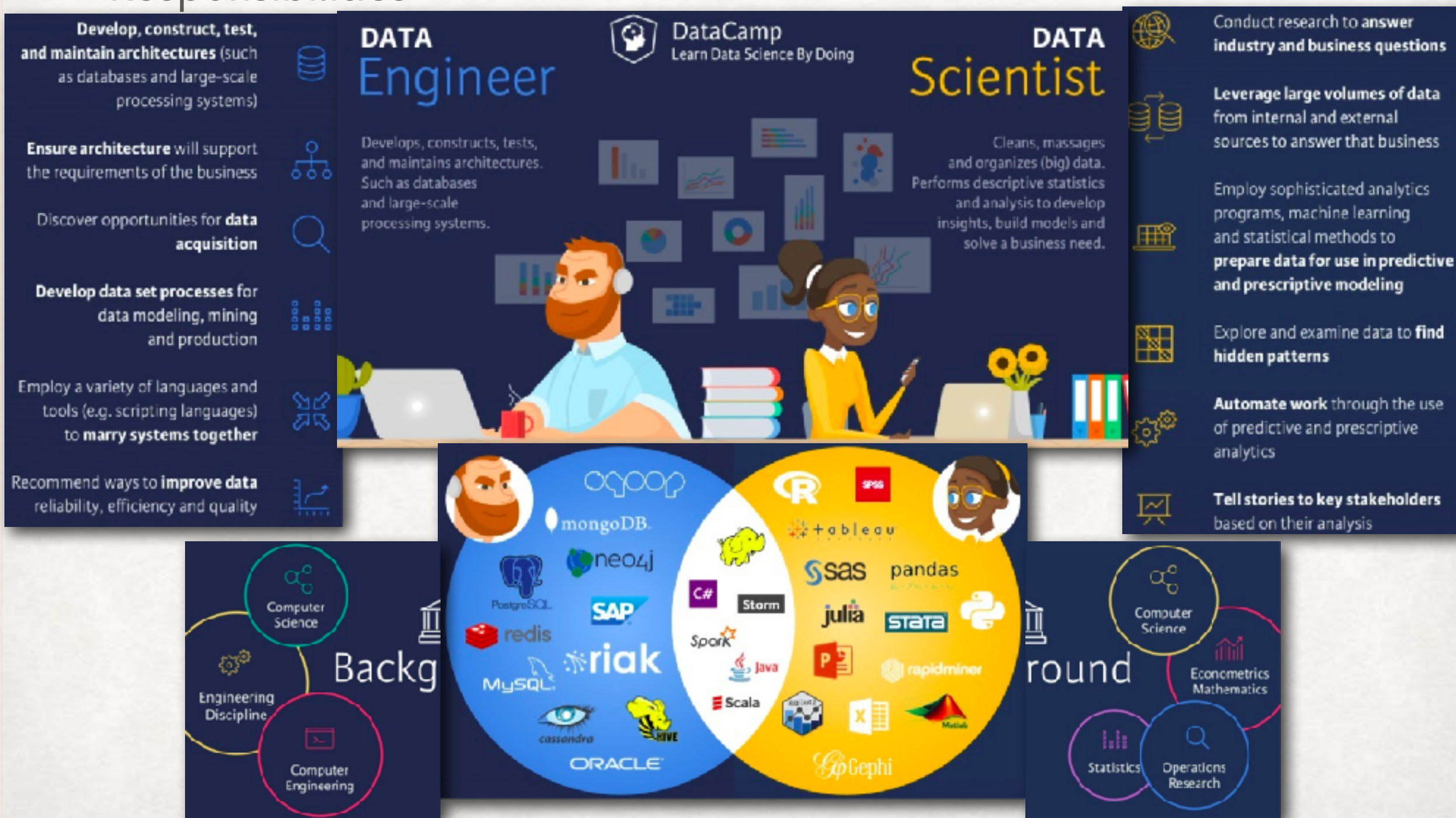
大綱

- 資料工程師與資料科學家
- 實際上的機器學習開發
- 資料的介紹與評估(四大關鍵)
- 機器學習專案開發流程
- 什麼是機器學習
- 數據結構分類
- 探索式數據分析

DATA SCIENTIST VS DATA ENGINEER

The differences between data engineers and data scientists

- Responsibilities

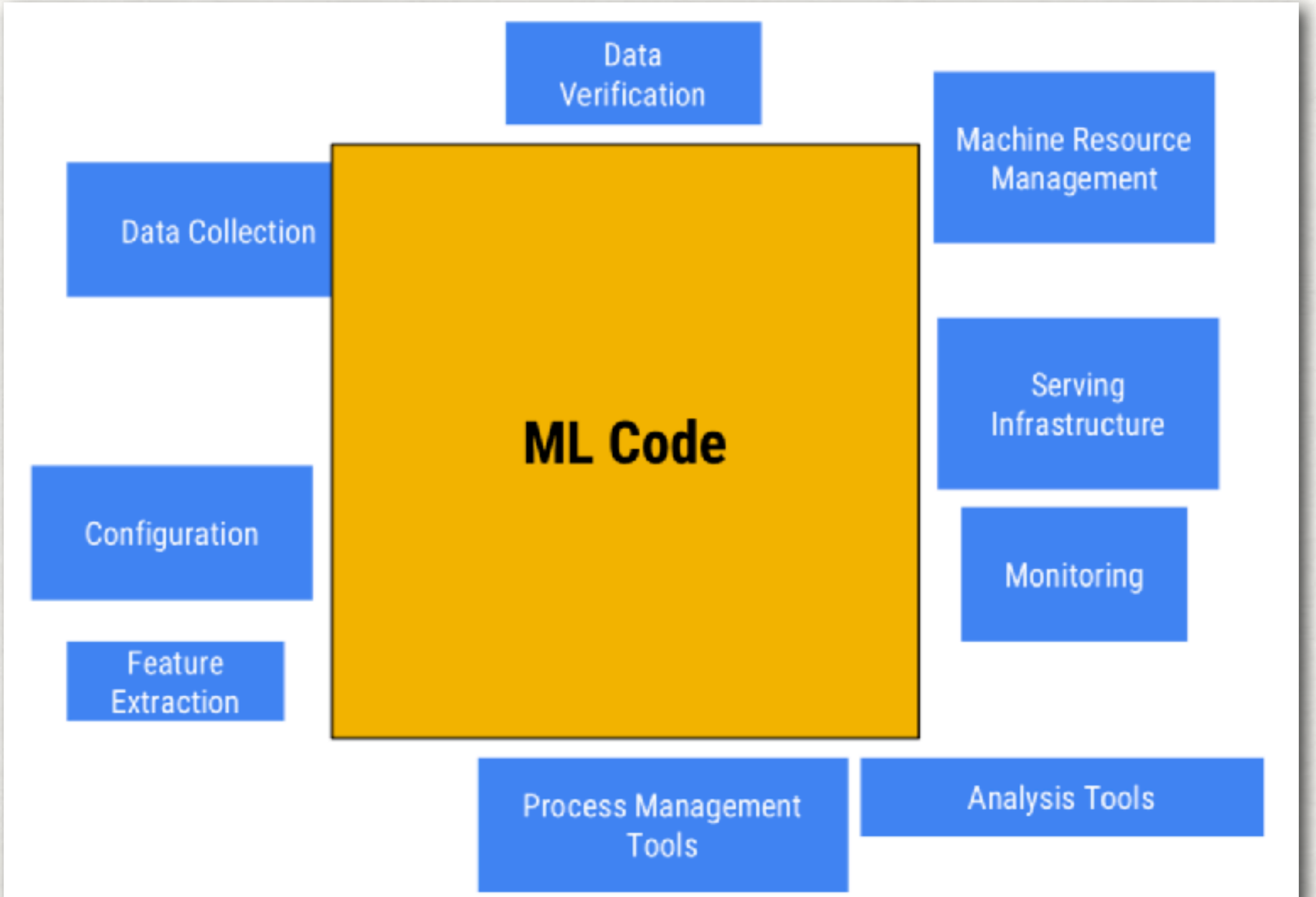


DATA SCIENTIST VS DATA ENGINEER

Salaries & Hiring

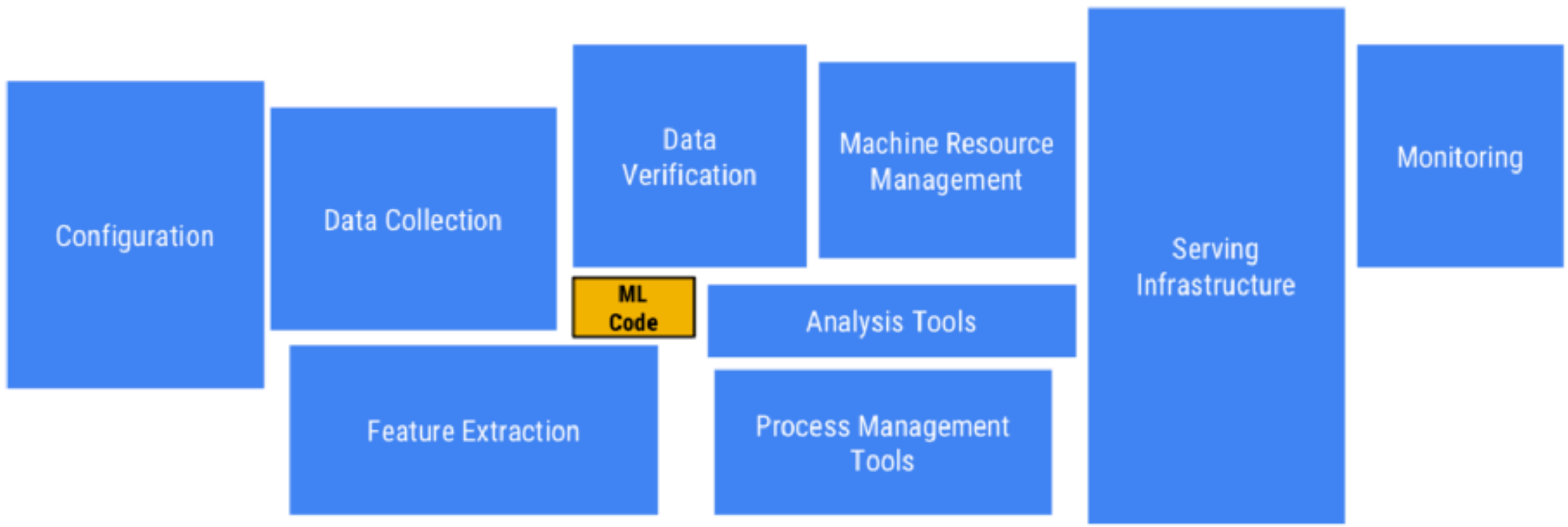
DATA SCIENTIST

我以為的資料科學家



DATA SCIENTIST

實際上的資料科學家



Source: Sculley et al.: Hidden Technical Debt in Machine Learning Systems

機器學習

鐵達尼號乘客生存資料

titanic_train

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	345763	18		S
20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7.225		C

機器學習

資料的意義？

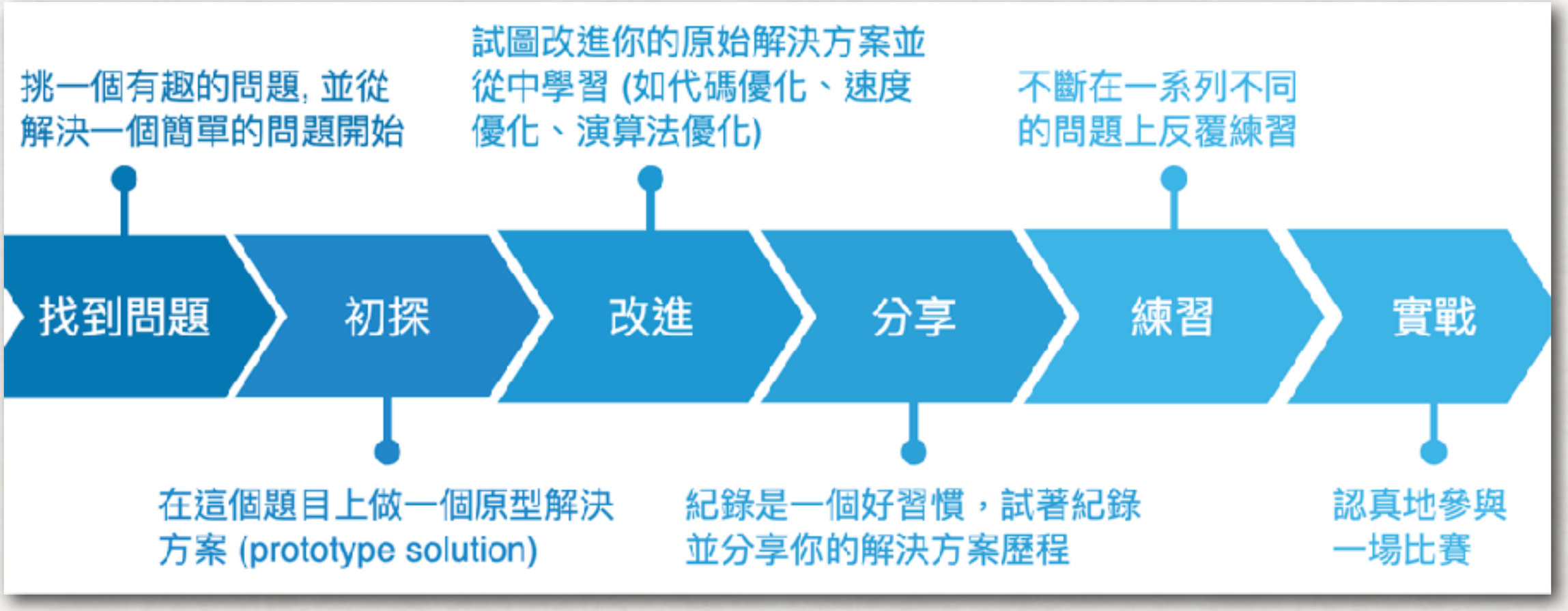
HomeCredit_columns_description

	Table	Row	Description
1	application_{train|test}.csv	SK_ID_CURR	ID of loan in our sample
2	application_{train|test}.csv	TARGET	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installmen
3	application_{train|test}.csv	NAME_CONTRACT_TYPE	Identification if loan is cash or revolving
4	application_{train|test}.csv	CODE_GENDER	Gender of the client
5	application_{train|test}.csv	FLAG_OWN_CAR	Flag if the client owns a car
6	application_{train|test}.csv	FLAG_OWN_REALTY	Flag if client owns a house or flat
7	application_{train|test}.csv	CNT_CHILDREN	Number of children the client has
8	application_{train|test}.csv	AMT_INCOME_TOTAL	Income of the client
9	application_{train|test}.csv	AMT_CREDIT	Credit amount of the loan
10	application_{train|test}.csv	AMT_ANNUITY	Loan annuity
11	application_{train|test}.csv	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given
12	application_{train|test}.csv	NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan
13	application_{train|test}.csv	NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,...)
14	application_{train|test}.csv	NAME_EDUCATION_TYPE	Level of highest education the client achieved
15	application_{train|test}.csv	NAME_FAMILY_STATUS	Family status of the client
16	application_{train|test}.csv	NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)
17	application_{train|test}.csv	REGION_POPULATION_RELATIVE	Normalized population of region where client lives (higher number means the client lives in more populated region)
18	application_{train|test}.csv	DAYS_BIRTH	Client's age in days at the time of application
19	application_{train|test}.csv	DAYS_EMPLOYED	How many days before the application the person started current employment
20	application_{train|test}.csv	DAYS_REGISTRATION	How many days before the application did client change his registration
21	application_{train|test}.csv	DAYS_ID_PUBLISH	How many days before the application did client change the identity document with which he applied for the loan
22	application_{train|test}.csv	OWN_CAR_AGE	Age of client's car
23	application_{train|test}.csv	FLAG_MOBIL	Did client provide mobile phone (1=YES, 0=NO)
24	application_{train|test}.csv	FLAG_EMP_PHONE	Did client provide work phone (1=YES, 0=NO)
25	application_{train|test}.csv	FLAG_WORK_PHONE	Did client provide home phone (1=YES, 0=NO)
26	application_{train|test}.csv	FLAG_CONT_MOBILE	Was mobile phone reachable (1=YES, 0=NO)
27	application_{train|test}.csv	FLAG_PHONE	Did client provide home phone (1=YES, 0=NO)
28	application_{train|test}.csv	FLAG_EMAIL	Did client provide email (1=YES, 0=NO)
29	application_{train|test}.csv	OCCUPATION_TYPE	What kind of occupation does the client have
30	application_{train|test}.csv	CNT_FAM_MEMBERS	How many family members does client have
31	application_{train|test}.csv	REGION_RATING_CLIENT	Our rating of the region where client lives (1,2,3)
32	application_{train|test}.csv	REGION_RATING_CLIENT_W_CITY	Our rating of the region where client lives with taking city into account (1,2,3)
33	application_{train|test}.csv	WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for the loan
34	application_{train|test}.csv	HOUR_APPR_PROCESS_START	Approximately at what hour did the client apply for the loan
35	application_{train|test}.csv	REG_REGION_NOT_LIVE_REGION	Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)
36	application_{train|test}.csv	REG_REGION_NOT_WORK_REGION	Flag if client's permanent address does not match work address (1=different, 0=same, at region level)
37	application_{train|test}.csv	LIVE_REGION_NOT_WORK_REGION	Flag if client's contact address does not match work address (1=different, 0=same, at region level)
38	application_{train|test}.csv	REG_CITY_NOT_LIVE_CITY	Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)

機器學習

學習路徑

- 初入資料科學的探索流程
- 找到問題 → 初探 → 改進 → 分享 → 練習 → 實戰



資料介紹與評估資料

探索與思考的關鍵點

四大關鍵問題

- 為什麼這個問題重要
- 資料從何而來
- 資料的型態是什麼
- 回答問題的關鍵指標是什麼

Questions	Explanation	Examples
為什麼這個問題重要？ (Why it is important)	A. 好玩 B. 企業的核心問題 C. 公眾利益 / 影響政策方向 D. 對世界很有貢獻	A. 預測生存 (吃雞) 遊戲誰可以活得久, PUBG B. 用戶廣告投放, ADPC C. 停車方針, 計程車載客優化 D. 肺炎偵測
資料從何而來？ (Where do data come from)	• 來源與品質息息相關 • 根據不同資料源，我們可以合理的推測/懷疑異常資料異常的理由與頻率	資料來源如： 網站流量、購物車紀錄、網路爬蟲、格式化表單、Crowdsourcing、紙本轉電子檔
資料的型態是什麼？ (What are they)	A. 結構化資料需要檢視欄位意義以及名稱 B. 非結構化資料需要思考資料轉換與標準化方式	A. 結構化：數值, 表格, ...etc B. 非結構化：圖像、影片、文字、音訊, ... etc
我們可以回答什麼問題 ？ 問題：指標 (What is our goal)	每個問題都應該要可以被驗證 → 有一個可供衡量的數學評估指標 (Evaluation Metrics)	常見的衡量指標如： 分類問題：正確率, AUC, MAP, ...etc 迴歸問題：MAE, RMSE, ...etc 補充資料：衡量指標

資料介紹與評估資料

練習一

一. 試著在Kaggle,上找一組競賽或資料寫下：

1. 你選的這組資料為何重要

2. 資料從何而來 (tips: 譬如提供者是誰、以什麼方式蒐集)

3. 蒐集而來的資料型態為何

4. 這組資料想解決的問題如何評估

二. 請寫一個函式用來計算 Mean Square Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

什麼是機器學習？

機器學習簡介 — 定義和組成

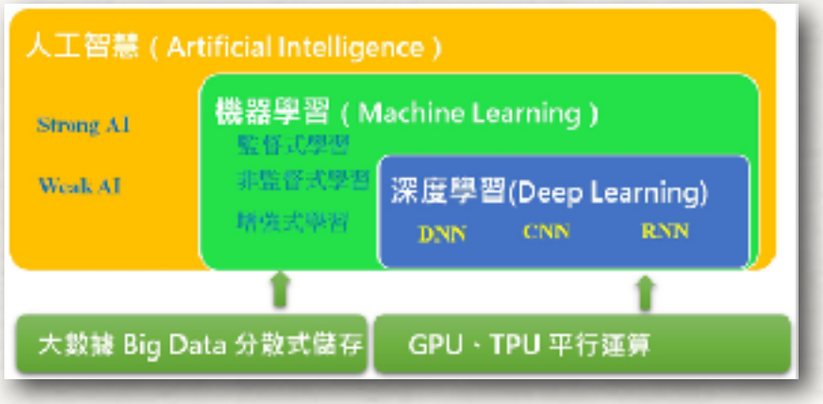
- Tom M.Mitchell：A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

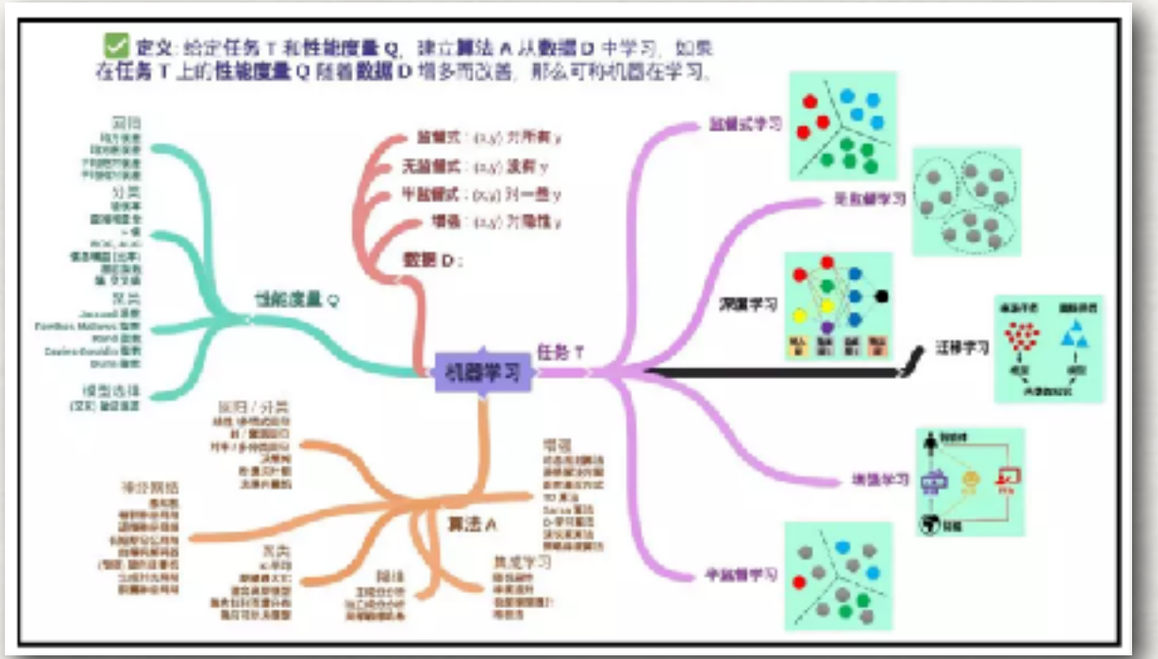
- 上面這段定義了機器學習包含的四元素：
  數據(Data)、任務(Task)、
  性能度量(Quality Metric)、模型(Model)

- 將其英譯中：假設用性能度量 P 來評估機器在某類任務 T 的性能，若該機器通利用經驗 E 在任務 T 中改善其性能 P，那麼可以說機器對經驗 E 進行了學習。

- 數據(Data)是經驗的另外一種說法。

方法分類



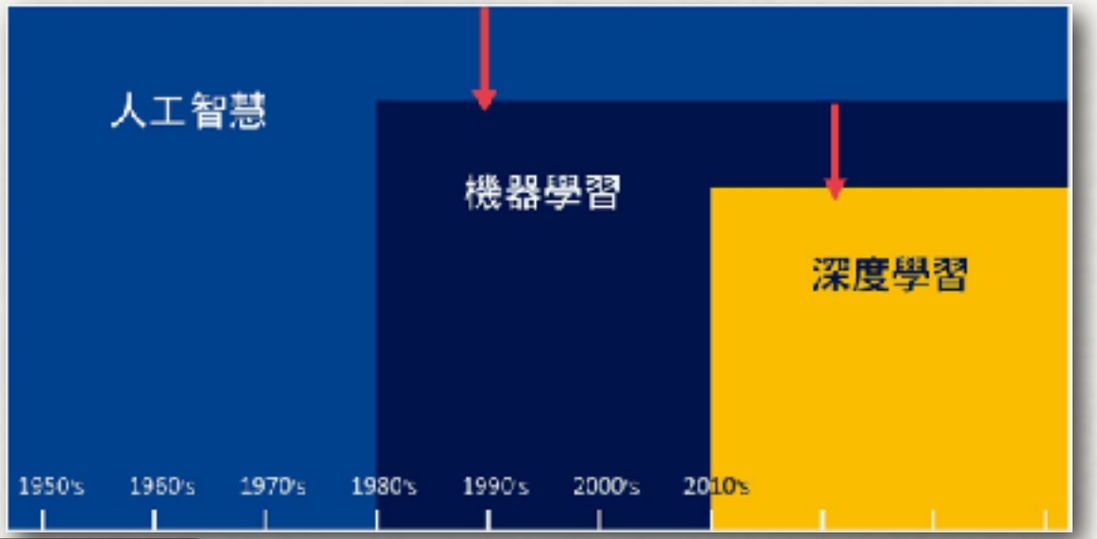


什麼是機器學習？

機器學習簡介

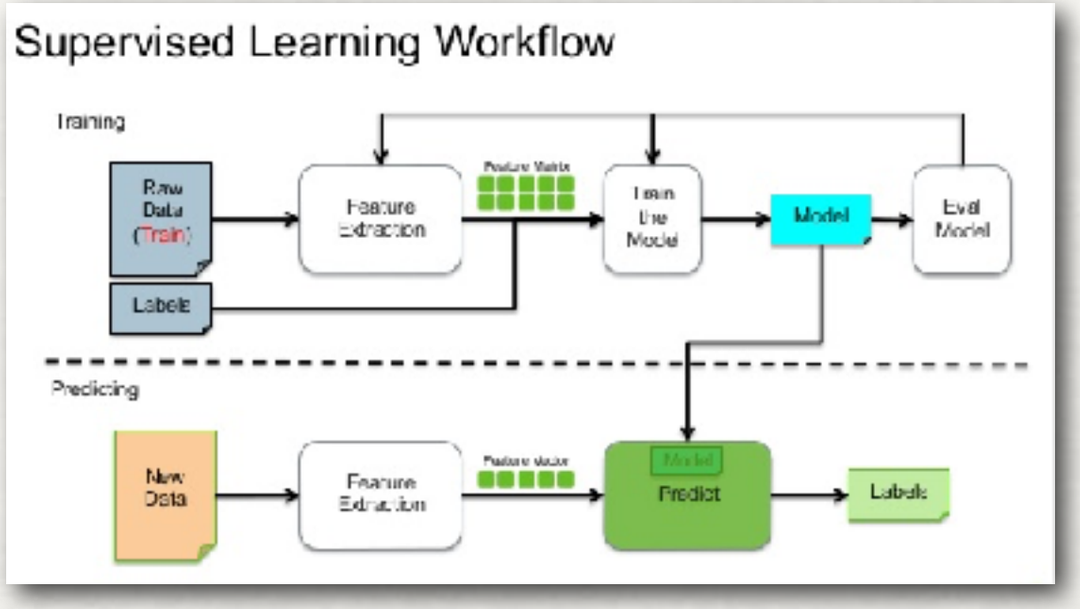
- 蒐集資料來分析與預測。
- 即使無法去完全地判斷出問題的本質模型，我們還是需要去建構一個近似模型。
- 機器學習不是去處理資料庫問題，而是人工智慧學習的問題。
- 機器學習使用資料或是過去的經驗將找到問題的最佳化效能。
- 機器學習找到的問題模型可以是預測性或是敘述性。
- 機器學習的演算法是有效率的。

人工智慧進程



建模流程

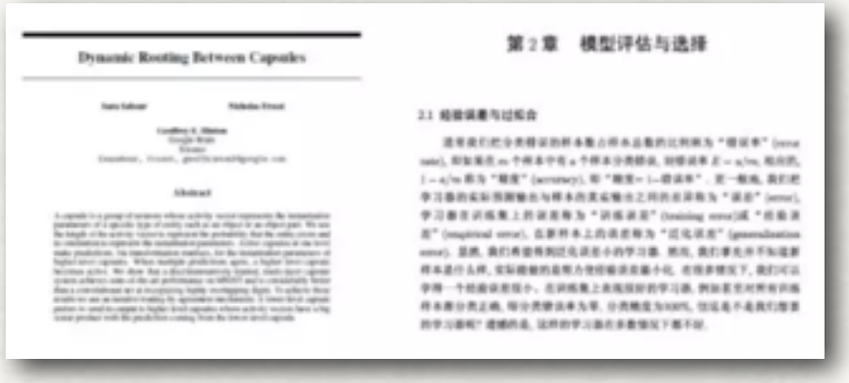
Data > Clean > Model > Predict



機器學習簡介

數據(*Data*) — 非結構化數據

- 數據(Data)是信息的載體，經驗的另一種說法。可以分為：
  - 數據類型 — 結構化數據和非結構化數據
  - 表達形式 — 原始數據和加工數據
  - 統計性質 — 樣本內數據和樣本外數據



文字

文本生成
閱讀理解

棋譜

AlphaGo

圖片

人臉辨識
物體分類

語音

語音辨識
機械對話

- 非結構化數據包含了文字、圖片、語音、影片和棋譜...等等。
- 這些非結構化數據經由深度學習(CNN、RNN)或增強式學習(AlphaGo)對數據學習和分類。

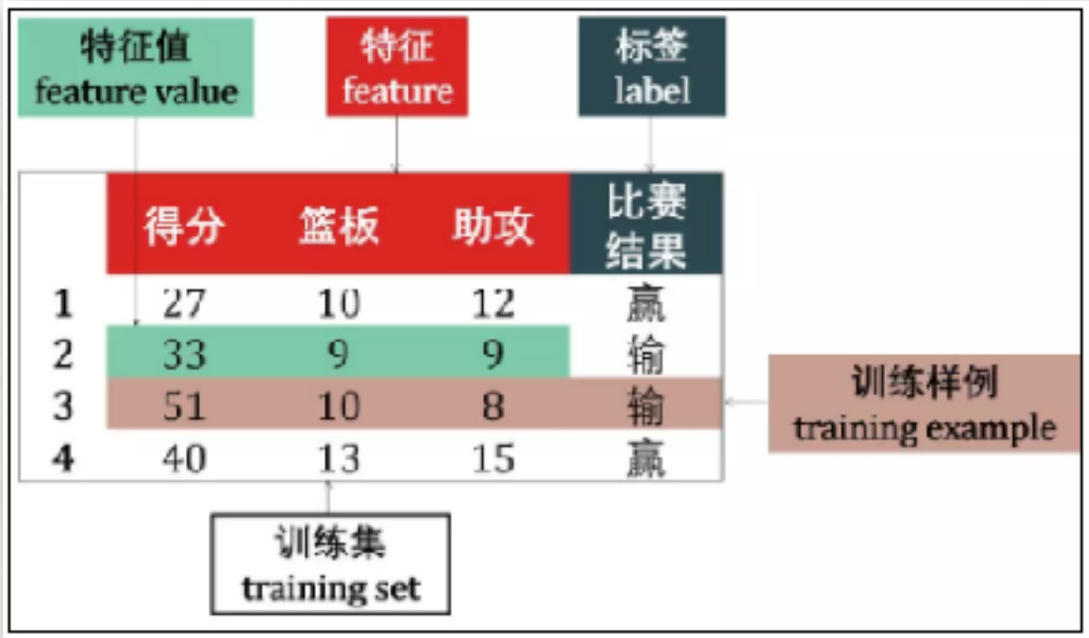
機器學習簡介

數據(Data) — 結構化數據

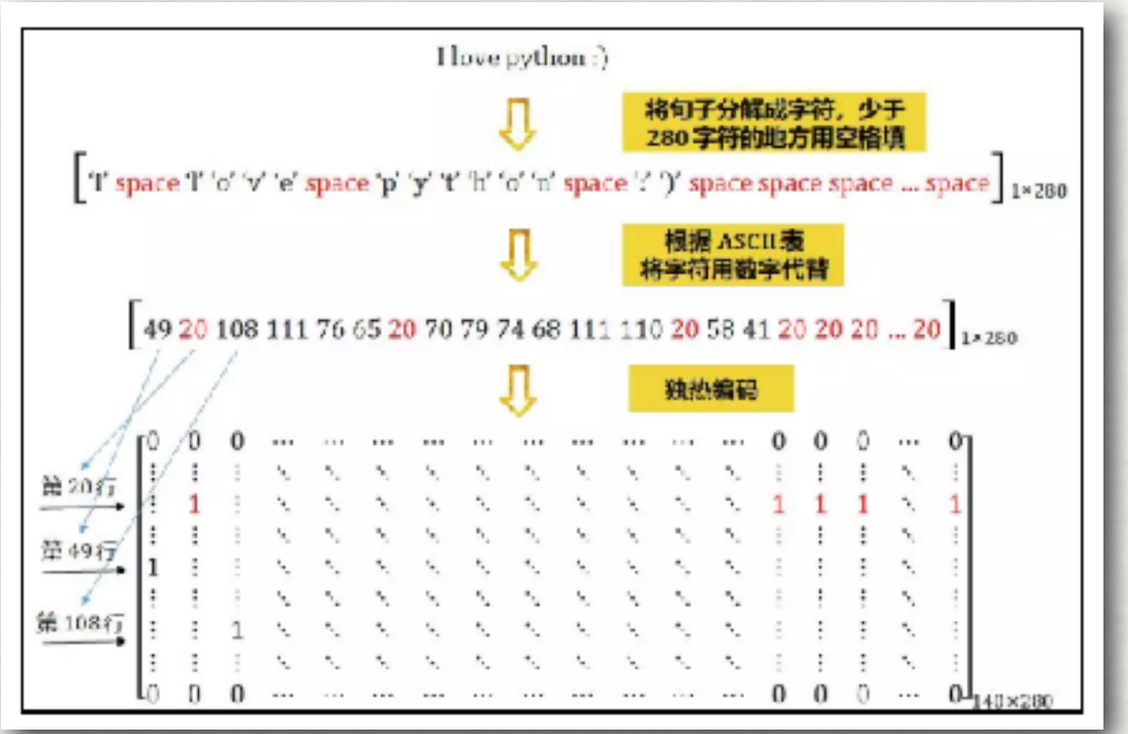
機器學習模型主要使用的是結構化數據，即二維的數據表。非結構化數據可以加工轉換成結構化數據，比如把

- 圖片類數據裡像素張量重塑成一維數組
- 文字類數據用獨熱編碼(one-hot encoding)轉成二維數組

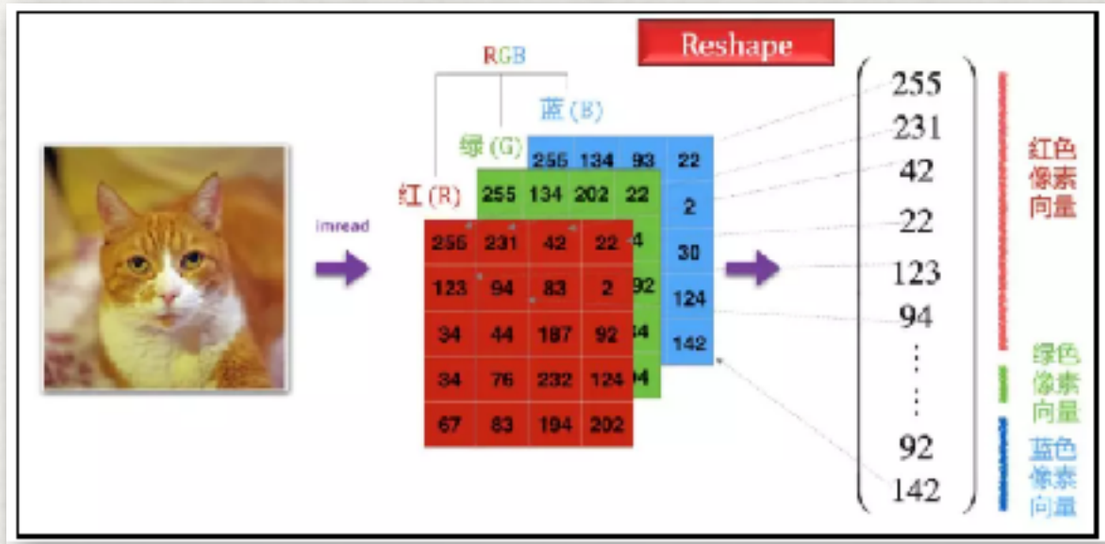
資料表結構



加工文字數據



加工圖片數據

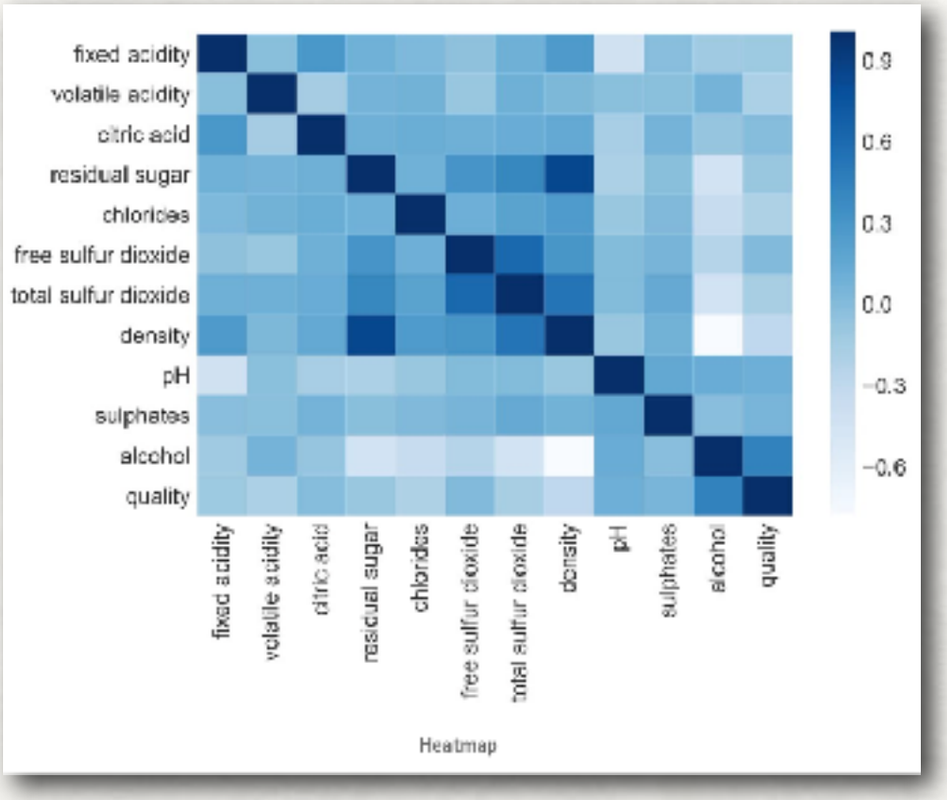


讀取資料：DATA SUMMARY

探索性數據分析

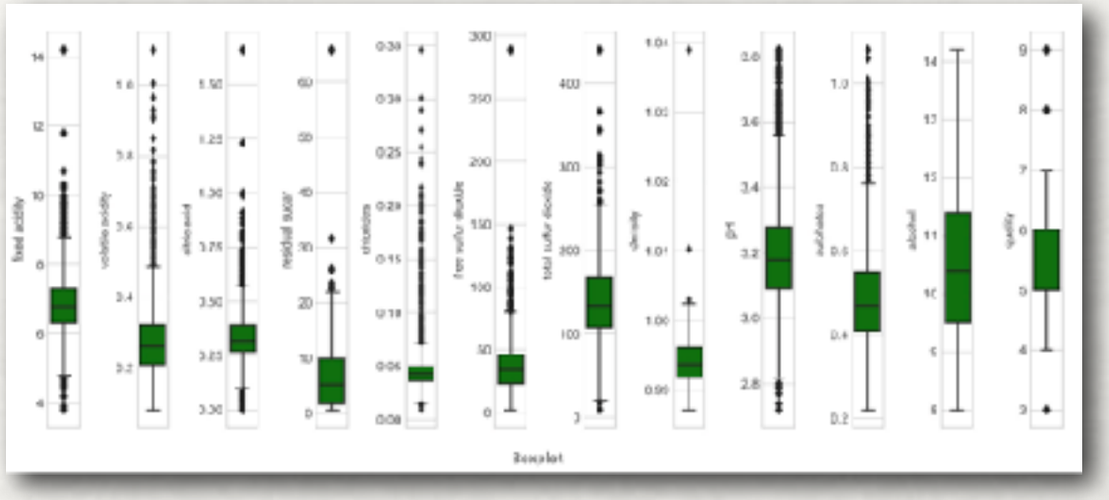
初步透過視覺化/統計工具進行分析，
達到三個主要目的：

- 了解資料
- 發現 outliers 或異常數值
- 分析各變數間的關聯性



常用函數

```
# 顯示dataframe前五筆資料
$ df.head()
# 顯示資料表的(rows, columns)
$ df.shape()
# 顯示資料描述
$ df.info()
# 顯示統計相關的資料
$ df.describe()
```



探索式資料分析

練習二

- 列出資料的大小（Hint：shape）
- 列出欄位
- 擷取部分資料（Hint：loc 或 iloc）
- …各種操作

PYTHON在AIOT的系統實作：機器學習

資料清理基礎

大綱

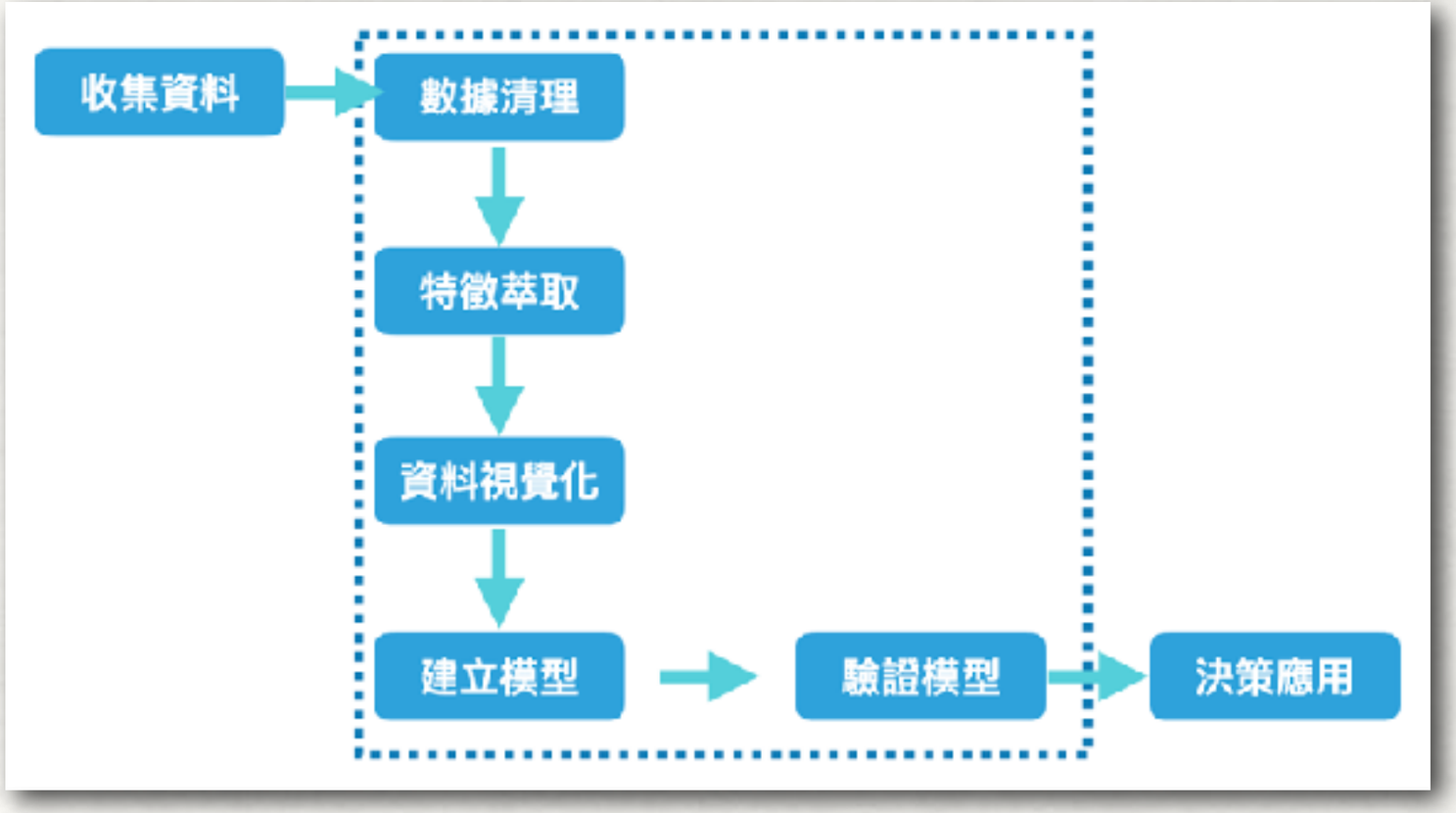
- 資料的意義和機器學習概論
- 建立DataFrame和讀取資料
- Outlier 和異常處理
- 常見的 DataFrame 操作
- 了解資料 Correlation
- 模型初體驗 Logistic Regression
- Plus：Sklearn 簡介

資料清理基礎

數據分析的流程

了解資料

- 資料說明
  - 多少個資料來源？資料的格式是什麼？
  - 資料之間關係是什麼？
  - 資料欄欄位的意義？每一個 row 的意義？



發現異常

- 檢查資料是否有誤

關聯性

- 找出重要的變數

建立DATAFRAME和讀取資料

練習三

一. 試著在Kaggle,上找一組競賽或資料寫下：

1. 你選的這組資料為何重要

2. 資料從何而來 (tips: 譬如提供者是誰、以什麼方式蒐集)

3. 蒐集而來的資料型態為何

4. 這組資料想解決的問題如何評估

二. 請寫一個函式用來計算 Mean Square Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

資料清理基礎

Outlier 和 處理

檢查異常

- 檢查異常值的方法
- 統計值：如平均數、標準差、中位數、分位數
- 畫圖：如直方方圖、盒圖、次數累積分布等

處理異常

- 處理異常值
- 取代補值：中位數、平均數等
- 另建欄位
- 整欄不用

CUPOY

1

異常值 (Outliers) 出現的可能原因

1.所以未知值，隨意填補 (約定俗成的代入)

如年齡 = -1 或 999, 電話是 0900-123-456

2.可能的錯誤紀錄/手誤/系統性錯誤

如某本書在某筆訂單的銷售量 = 1000 本

2

檢查 Outliers 的流程與方法

- 盡可能確認每一個欄位的意義 (但有些競賽資料不會提供欄位意義)
- 透過檢查數值範圍 (五值、平均數及標準差) 或繪製散點圖 (scatter)、分布圖 (histogram) 或其他圖檢查是否有異常。

3

對 Outliers 的處理方法

- 新增欄位用以紀錄異常與否
- 填補 (取代)
- 視情況以中位數, Min, Max 或平均數填補(有時會用 NA)

OUTLIER 和 處理

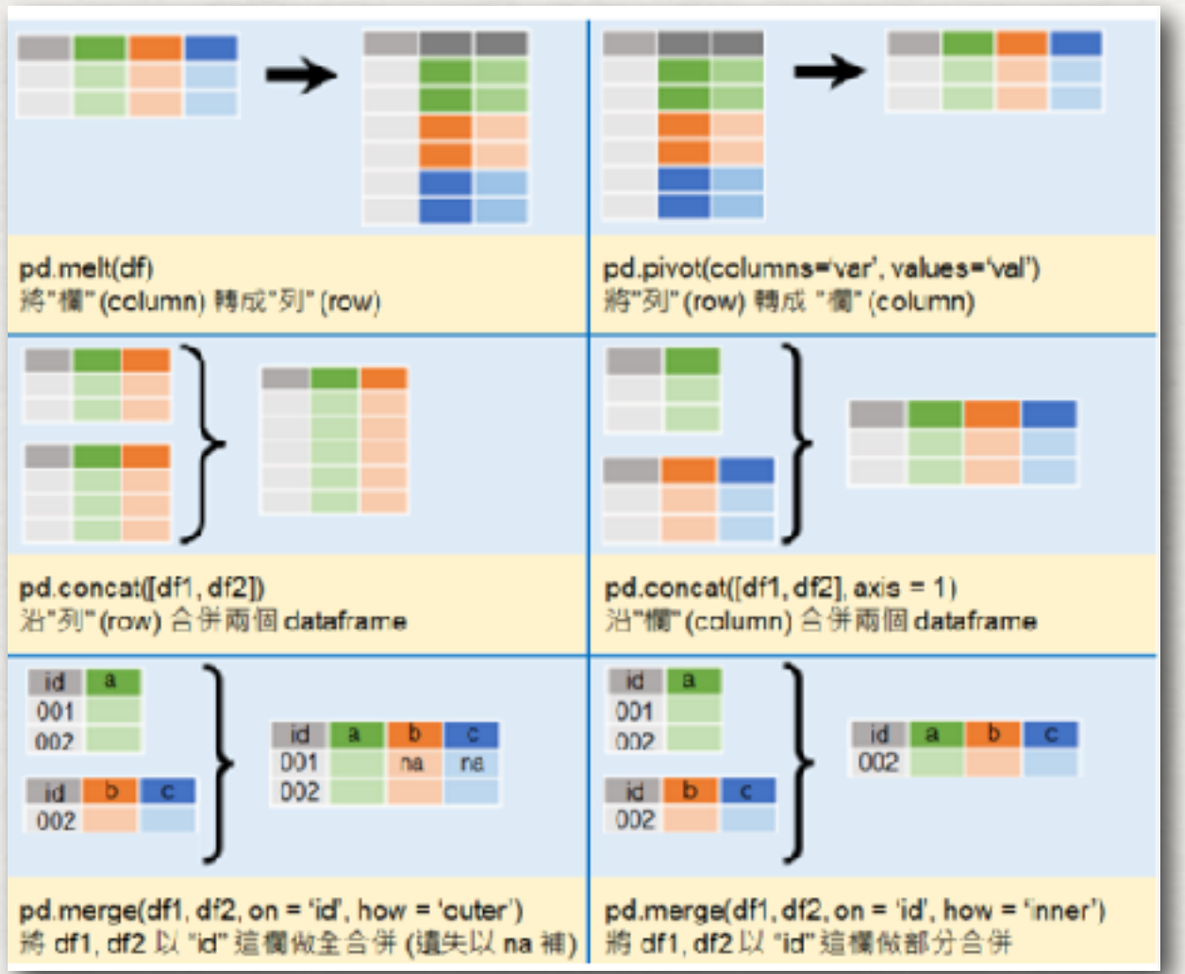
練習四

- 找出常見(不可能發生的)錯誤，年齡 0 , 999
- 找出系統錯誤 / 手誤
- 利用直方圖檢視異常值
- 找出不合理的數值 (並標記由模型來解釋)

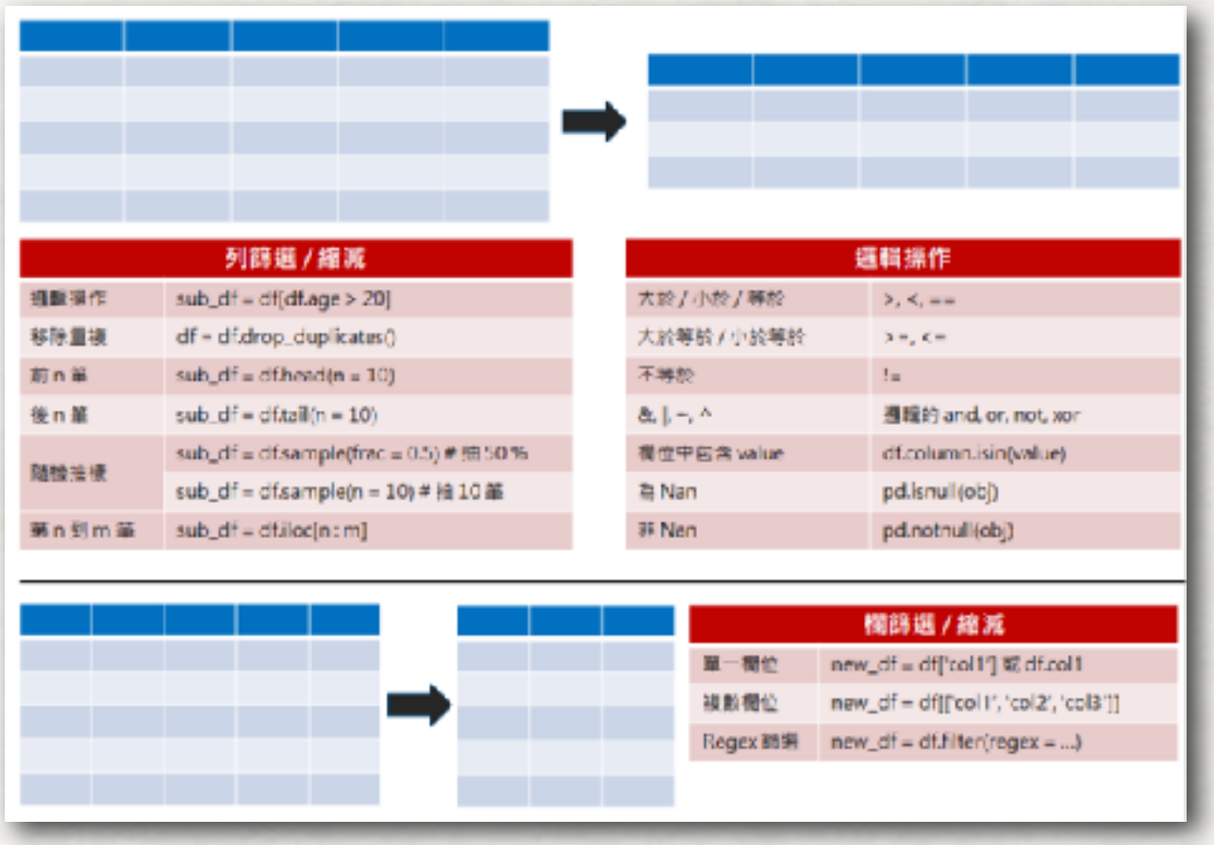
資料清理基礎

常用的 *DataFrame* 操作

轉換與合併



SUBSET



列篩選 / 縮減	
邏輯操作	sub_df = df[df.age > 20]
移除重複	df = df.drop_duplicates()
前 n 筆	sub_df = df.head(n = 10)
後 n 筆	sub_df = df.tail(n = 10)
隨機抽樣	sub_df = df.sample(frac = 0.5) # 抽 50%
	sub_df = df.sample(n = 10) # 抽 10 筆
第 n 到 m 筆	sub_df = df.iloc[n : m]

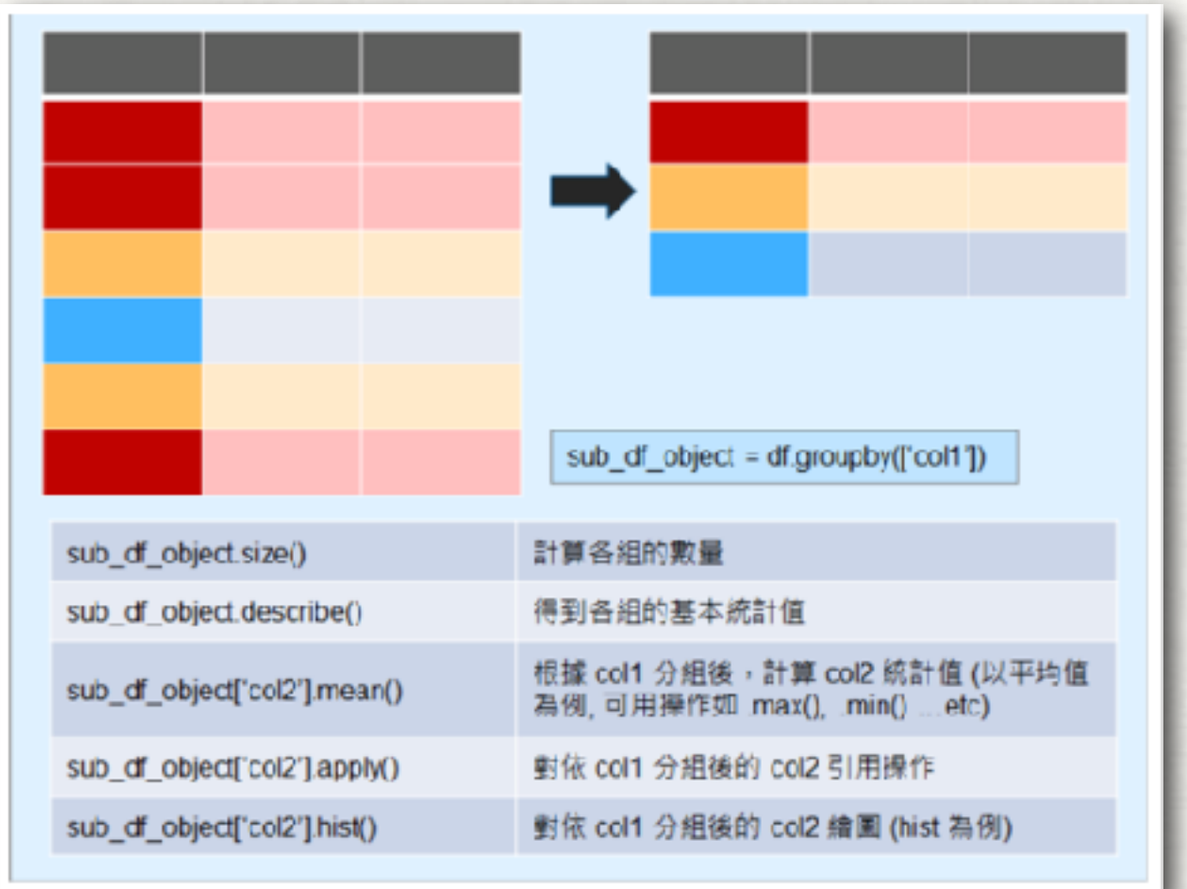
邏輯操作	
大於 / 小於 / 等於	>, <, ==
大於等於 / 小於等於	>=, <=
不等於	!=
&, |, ~, ^	邏輯的 and, or, not, xor
欄位中包含 value	df.column.isin(value)
為 Nan	pd.isnull(obj)
非 Nan	pd.notnull(obj)

欄篩選 / 縮減	
單一欄位	new_df = df['col1'] 或 df.col1
複數欄位	new_df = df[['col1', 'col2', 'col3']]
Regex 篩選	new_df = df.filter(regex = ...)

資料清理基礎

常用的 *DataFrame* 操作

GROUP



sub_df_object.size()	計算各組的數量
sub_df_object.describe()	得到各組的基本統計值
sub_df_object['col2'].mean()	根據 col1 分組後，計算 col2 統計值 (以平均值為例, 可用操作如 max(), min() ...etc)
sub_df_object['col2'].apply()	對依 col1 分組後的 col2 引用操作
sub_df_object['col2'].hist()	對依 col1 分組後的 col2 繪圖 (hist 為例)

重點複習

- 合併 (concat) 常用於將多個表依照某欄(key) 結合使用
- 分組 (groupby) 是常用在計算 "組"統計值時會用到的功能
- 許多基本操作 (如 >, ==, <, ~) 都是可以在pandas 作為篩選條件使用

常用的 DATAFRAME 操作

練習五

- 使用 merge / transform , subset , groupby 等操作
- 請將 app_train 中的 CNT_CHILDREN 依照下列規則分為四組，並將其結果在原本的 dataframe 命名CNT_CHILDREN_GROUP
- 請根據 CNT_CHILDREN_GROUP 以及 TARGET，列出各組的平均 AMT_INCOME_TOTAL，並繪製 baxplot
- 請根據 CNT_CHILDREN_GROUP 以及 TARGET，對 AMT_INCOME_TOTAL 計算 Z 轉換 後的分數

資料清理基礎

了解資料相關係數

相關係數是其中一個常用來了解各個欄位與我們想要預測的目標之間的關係的指標。相關係數衡量兩個**隨機變量**之間線性關係的強度和方向。雖然不是表示變數之間關係的最好方法，但可以提供我們很直觀的了解。

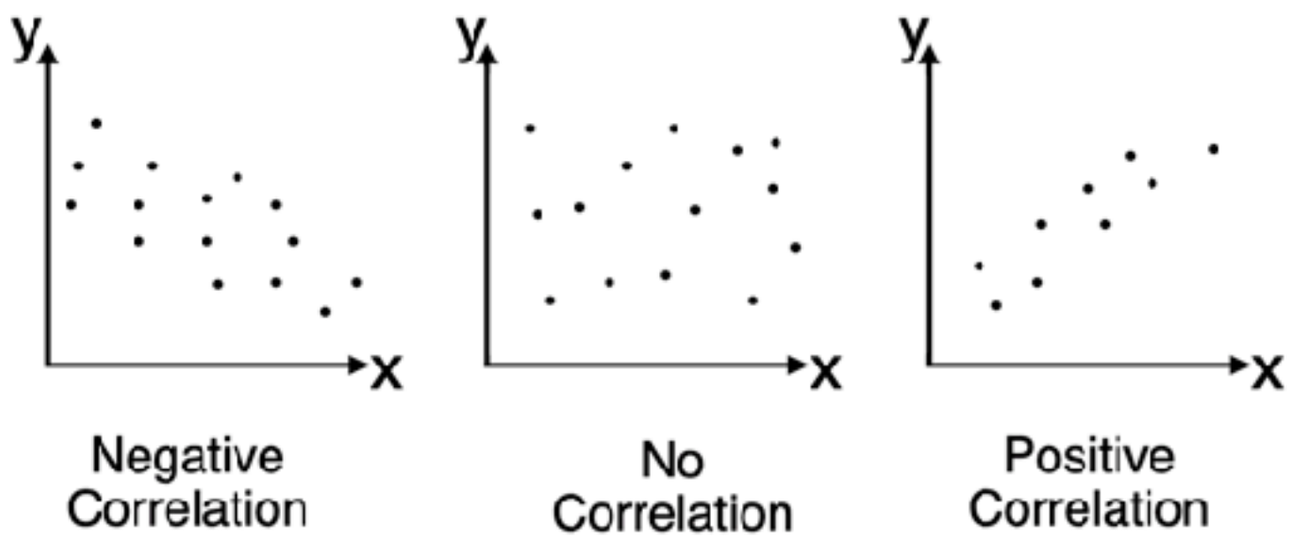
$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

資料清理基礎

了解資料相關係數

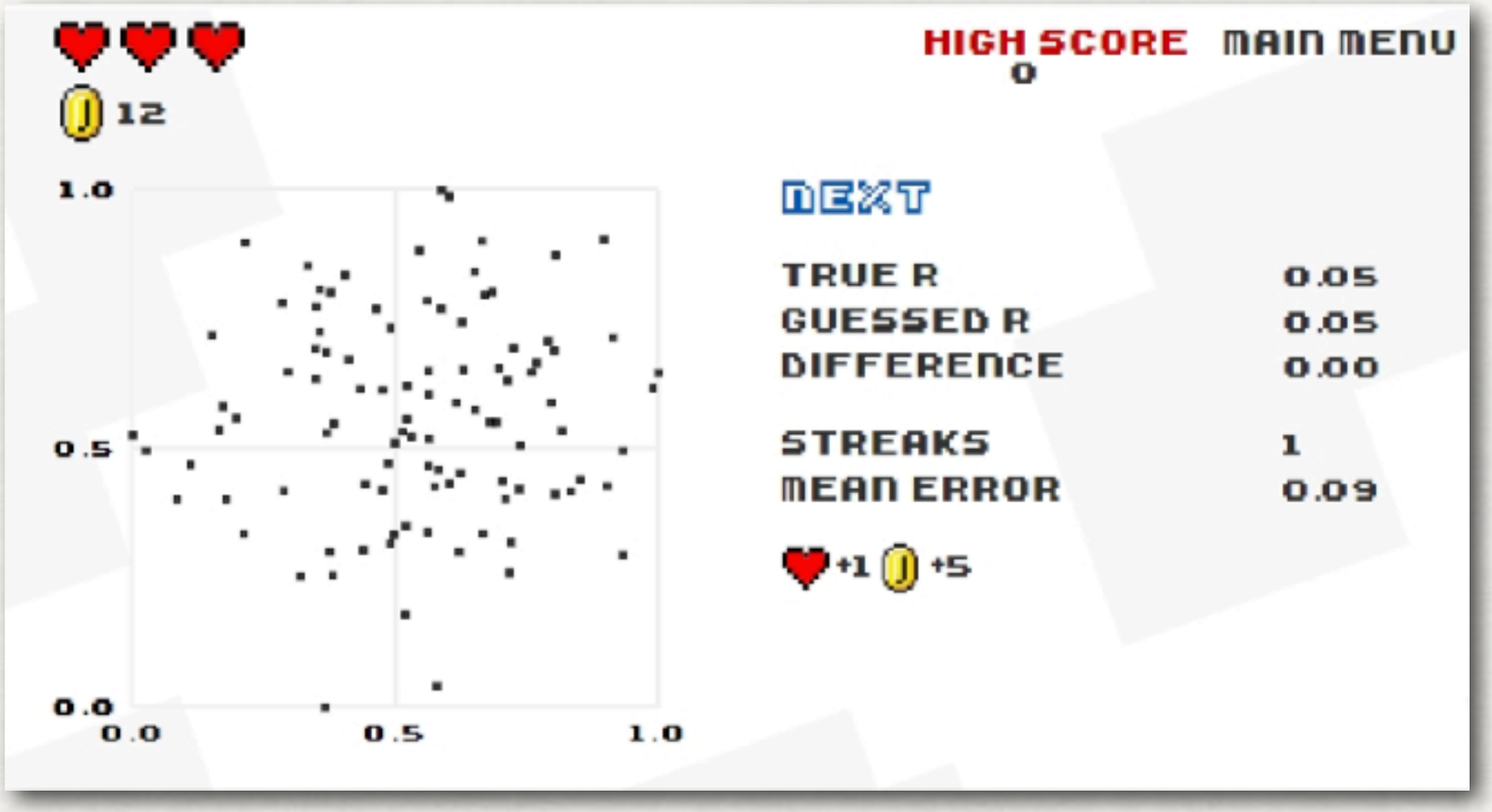
相關係數是一個介於 -1～1 之間的值，負值代表負相關，正值代表正相關，數值的大小代表相關性的強度

- .00-.19: 非常弱相關
- .20-.39: 弱相關
- .40-.59: 中度相關
- .60-.79: 強相關
- .80-1.0: 非常強相關



了解資料相關係數

練習五

- 相對係數小遊戲
- http://guessthecorrelation.com/
- 

資料清理基礎

模型初體驗 *Logistic Regression*

- 我們最終的目的是要預測客戶是否會違約遲繳貸款的機率，在我們開始使用用任何複雜的模型之前，有一個最簡單的模型當作baseline 是一個好習慣。

- 大部分的情況，如何通過資料得到模型的具體計算過程的程式碼是不需要我們自己的，python 有很豐富的資源，我們只要把相關的套件 import 進來，看官方文檔或者 google 怎麼使用就好。（通常只是一個 .fit 的過程）

- 相反地，時間會花更更多在讀取資料、處理資料上，甚至，第一次把模型結果儲存為對的格式可能比訓練模型還花時間。

我們使用 Logistic Regression 作為 baseline，理論的部分可以參考

- Andrew Ng's videos about logistic regression

  https://www.youtube.com/playlist?list=PLNeKWBMsAzboR8vvhnlanxCNr2V7ITuxy

模型初體驗 LOGISTIC REGRESSION

練習六

- 操作範例試著做出你的第一個模型！
-

PYTHON在AIOT的系統實作：機器學習

SKLEARN 簡介

大綱

- 機器學習簡介
- 建立模型四步驟
- 機器學習生態系
- Mission Module
- Pseudo code
- Sklearn 框架使用步驟
- Plus：Overfit , Penalty , Alpha

機器學習簡介

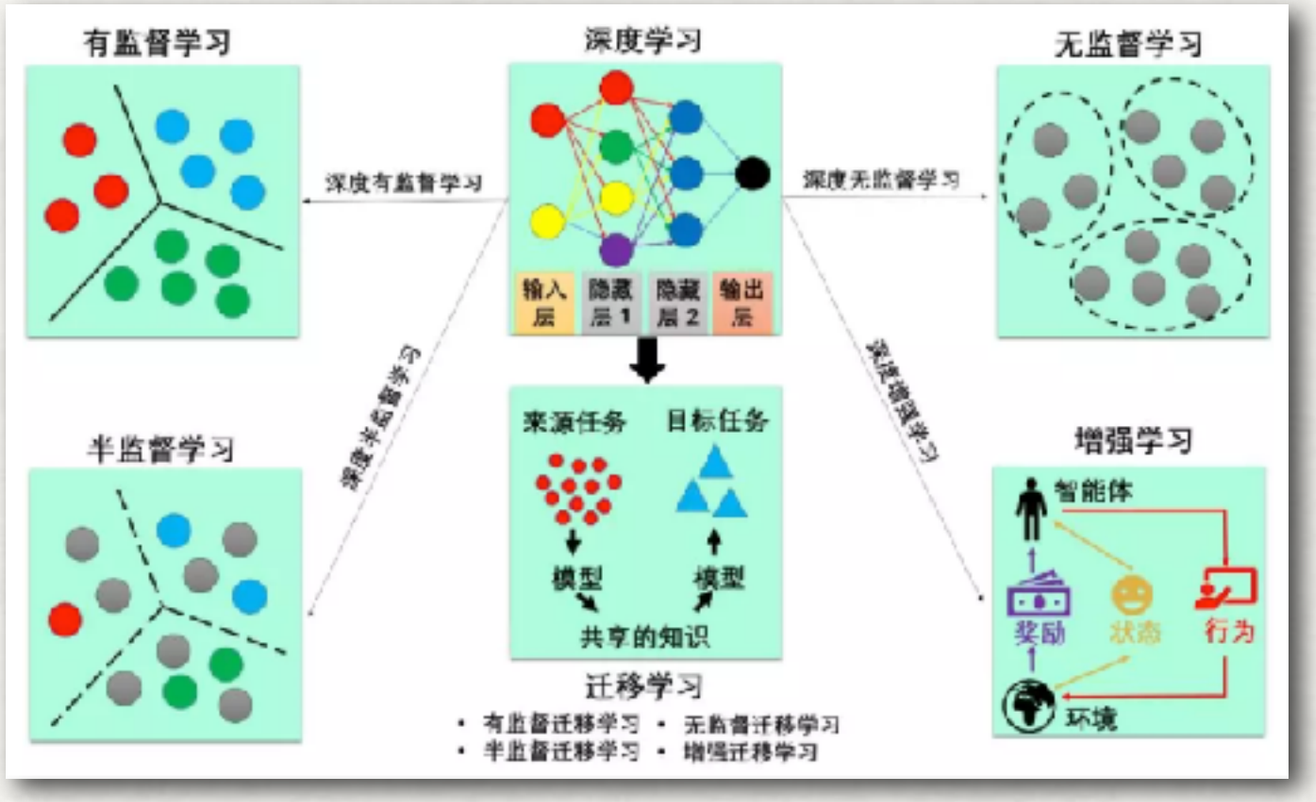
任務(*Task*)

任務分類

根據學習的任務模式 (訓練數據是否有標籤)，機器學習可分為四大類：

- 監督式學習(有標籤)
- 非監督學習(無標籤)
- 半監督學習(部分標籤)
- 增強學習(等級標籤)

模式與方法



分法分類

深度學習只是一種方法，而不是任務模式，因此與上面四類不屬於同一個維度，但是深度學習與它們可以疊加成：深度有監督學習、深度非監督學習、深度半監督學習和深度增強學習。

遷移學習也是一種方法，也可以分類為有監督遷移學習、非監督遷移學習、半監督遷移學習和增強遷移學習。

機器學習簡介

性能度量(*Quality Metric*)

回歸和分類任務中有一些最常見的損失函數以及一些有用的性能度量如下：

任务	性能度量	
	误差函数	其它度量
回归	均方、均方根、平均绝对/相对误差、Huber	-
分类	0-1、对数、指数、合页	错误率、查全率、查准率、F_1 得分、AUC、ROC

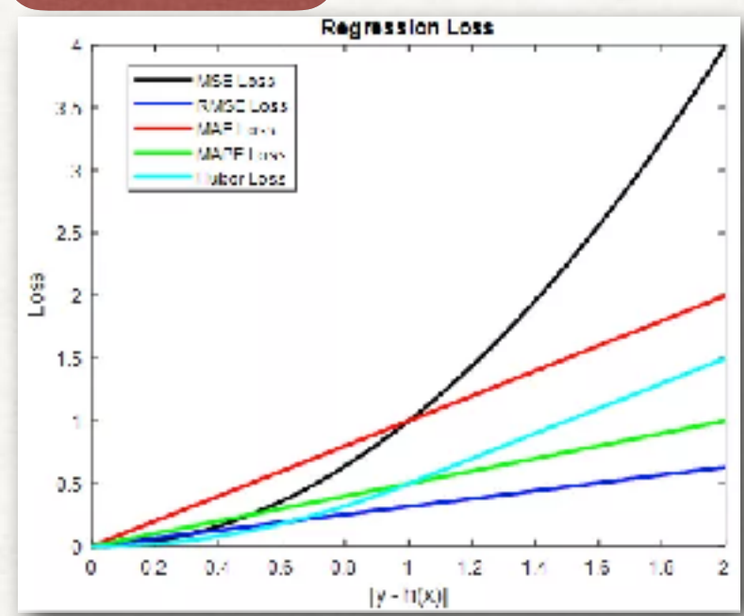
回歸問題
損失函數常用：

- 均方誤差(MSE)
- 均方根誤差(RMSE)
- 平均絕對誤差(MAE)
- 平均相對誤差(MAPE)
- Huber

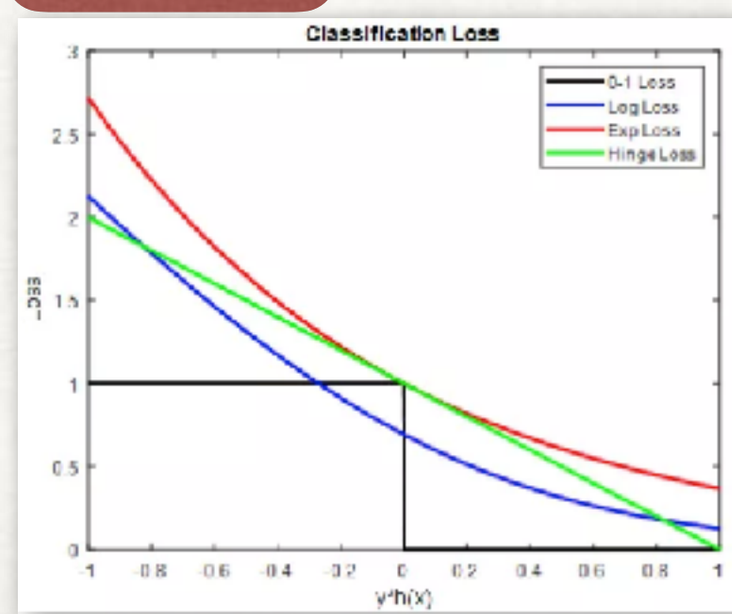
分類問題
損失函數常用：

- 錯誤率(error rate)
- 精準度(accuracy)
- Precision
- Recall
- F1 score

回歸任務



分類任務

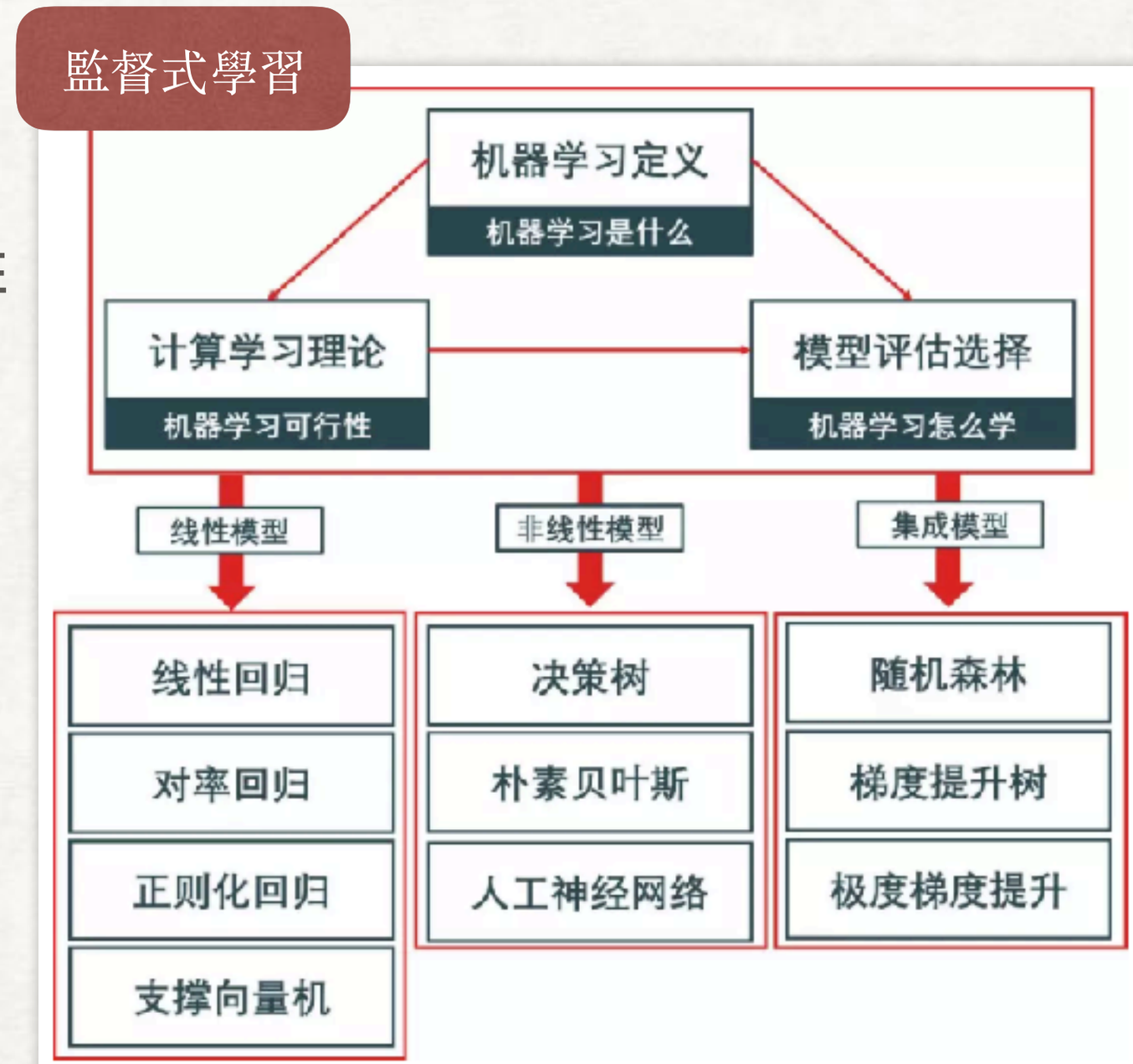


機器學習簡介

模型(*Model*)

- 監督式模型可以分為線性、非線性和集合模型，各回歸模型與SVM(向量機)都為線性模型，另外決策樹可以同時解決回歸和分類問題，右圖為監督式模型分類示意圖。

- 非監督式模型包括各種聚類分析 (KMeans, DBSCAN)、主成分分析 (PCA)、獨立成分分析 (ICA)、隱含狄利克雷分配 (LDA) 等等。



SKLEARN

建立模型四步驟

在 *Scikit-learn* 中，建立一個機器學習的模型其實非常簡單，流程大略是以下四個步驟

讀取資料

- 1. 讀進資料，並檢查資料的 shape (有多少 samples (rows), 多少 features (columns)，label 的型態是什麼？)
- 讀取資料的方法：
  使用 pandas 讀取 .csv 檔：pd.read_csv
  使用 numpy 讀取 .txt 檔：np.loadtxt
  使用 Scikit-learn 內建的資料集：
  sklearn.datasets.load_xxx
- 檢查資料數量：data.shape (data should be np.array or dataframe)

資料切分

- 2. 將資料切為訓練 (train) / 測試 (test)

  train_test_split(data)

建立模型

- 3. 建立模型，將資料 fit 進模型開始訓練

```
clf = DecisionTreeClassifier()
clf.fit(x_train, y_train)
```

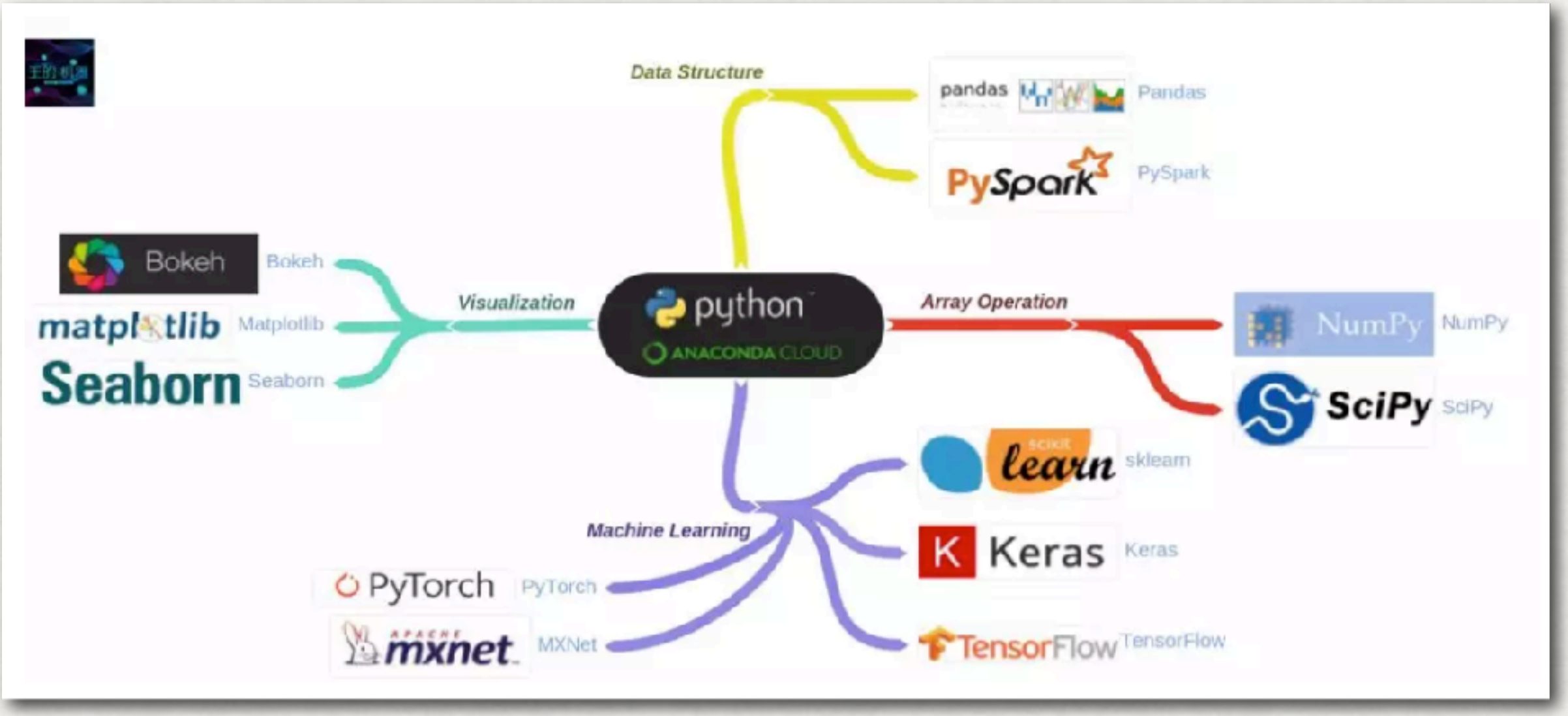
測試模型

- 4. 將測試資料 (features) 放進訓練好的模型中，得到 prediction，與測試資料的 label (y_test) 做評估

```
clf.predict(x_test)
accuracy_score(y_test, y_pred)
f1_score(y_test, y_pred)
```

PYTHON

Machine Learning Network



SKLEARN

Misson Module

- 分類
- 回歸
- 聚類
- 降維
- 模型選擇
- 預處理

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

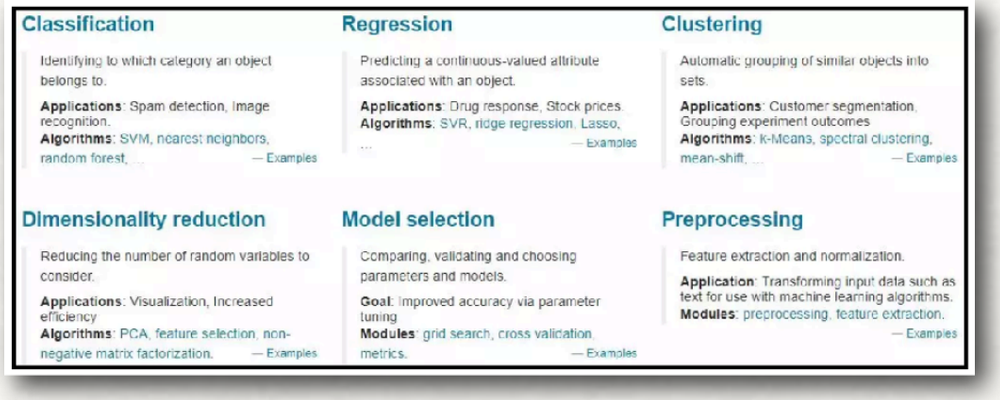
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

SKLEARN

Pseudo code

- 分類
- 回歸
- 聚類
- 降維
- 模型選擇
- 預處理



Classification

Identifying to which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

要使用上述六大模塊的方法，可以參考以下的Pseudo code(偽代碼)，注意 import 後面用的都是一些通用名稱，如 SomeClassifier, SomeRegressor, SomeModel，具體化的名稱由具體問題而定，比如：

- SomeClassifier = RandomForestClassifier
- SomeRegressor = LinearRegression
- SomeModel = KMeans, PCA
- SomeModel = GridSearchCV, OneHotEncoder

SKLEARN

Pseudo code

- SomeClassifier, SomeRegressor, SomeModel 其實都叫做估計器 (estimator)，就像 Python 裡「萬物皆物件」那樣，Sklearn 裡「萬物皆估計器」。
- 此外，Sklearn 裡面還有很多自帶數據集供，引入它們的偽代碼如下。

数据集 (Dataset)

```
from sklearn.datasets import SomeData
```

分类 (Classification)

```
from sklearn import SomeClassifier
from sklearn.linear_model import SomeClassifier
from sklearn.ensemble import SomeClassifier
```

回归 (Regression)

```
from sklearn import SomeRegressor
from sklearn.linear_model import SomeRegressor
from sklearn.ensemble import SomeRegressor
```

聚类 (Clustering)

```
from sklearn.cluster import SomeModel
```

降维 (Dimensionality Reduction)

```
from sklearn.decomposition import SomeModel
```

模型选择 (Model Selection)

```
from sklearn.model_selection import SomeModel
```

预处理 (Preprocessing)

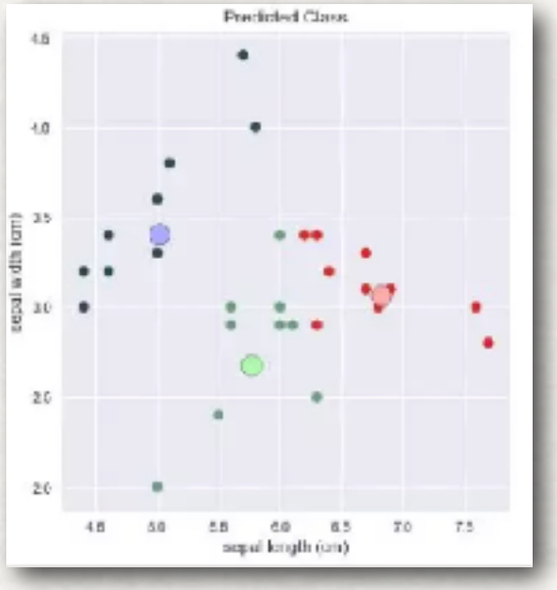
```
from sklearn.preprocessing import SomeModel
```

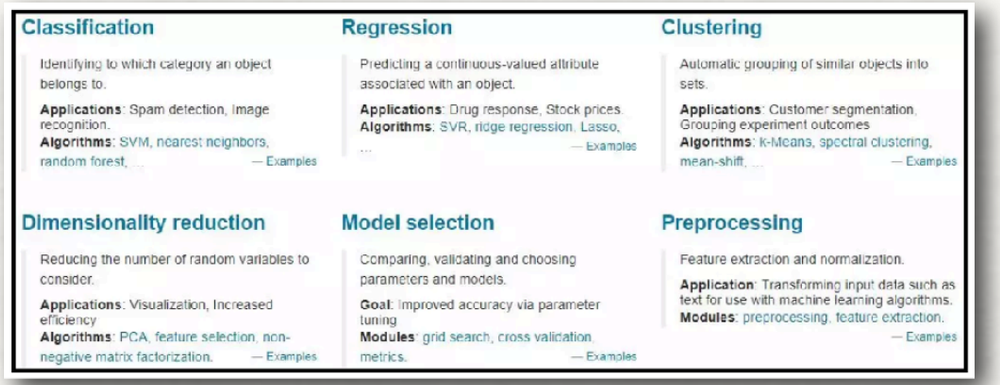
SKLEARN

四步驟使用*Sklearn*機器學習框架

確定任務

- 使用監督式回歸、分類方法？
- 使用分類或是降維？





Classification

Identifying to which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

數據預處理

- 處理缺失值、異常
- 編碼分類變數
- 正規和標準化

訓練和評估

- 訓練使用fit()
- 再使用predict()評估

選擇模型

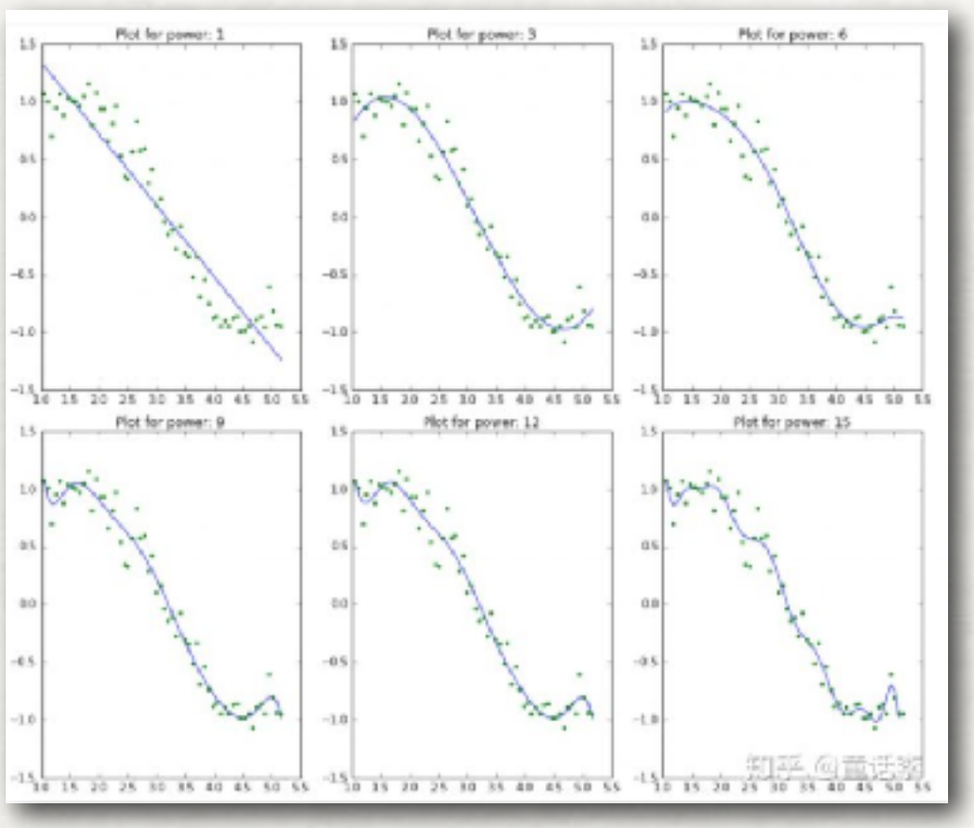
- 了解並且調整參數
- 選擇得分最高的模型

LINEAR REGRESSION

Over Fitting

$$y = x^0 + x^1 + \ldots + x^n$$

$$n = 1,\ldots,15$$



Power

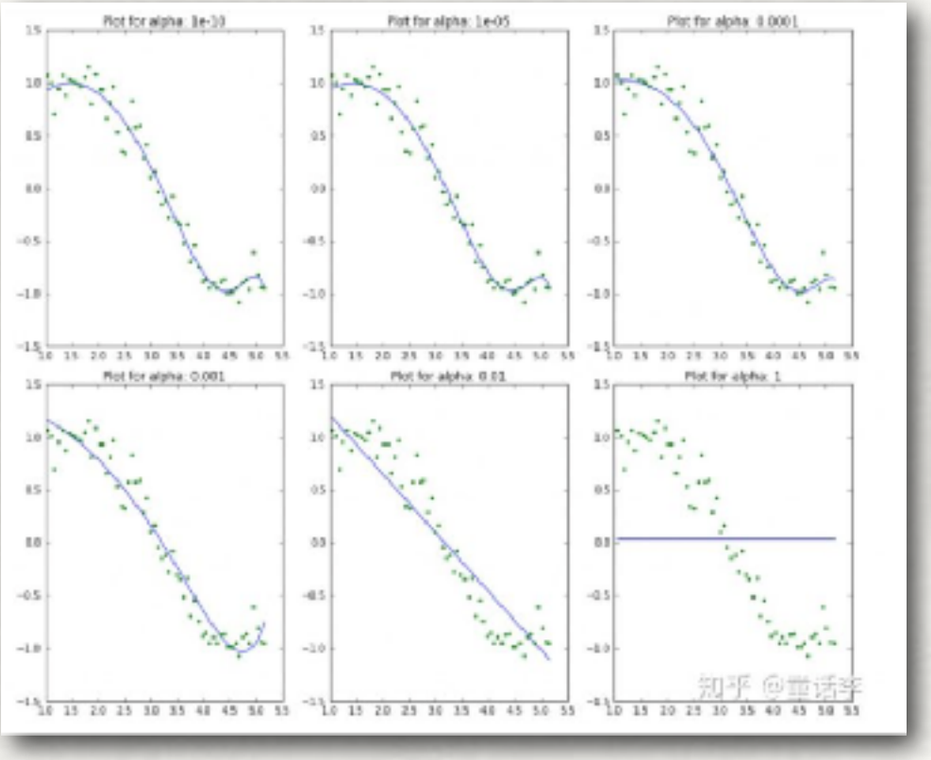
	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	c
model_pow_1	3.3	2	-0.62	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_2	3.3	1.9	-0.58	-0.006	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_3	1.1	-1.1	3	-1.3	0.14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_4	1.1	-0.27	1.7	-0.53	-0.036	0.014	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_5	1	3	-5.1	4.7	-1.9	0.33	-0.021	NaN	NaN	NaN	NaN	NaN	NaN	N
model_pow_6	0.99	-2.8	9.5	-9.7	5.2	-1.6	0.23	-0.014	NaN	NaN	NaN	NaN	NaN	N
model_pow_7	0.93	10	-56	69	-45	17	-3.5	0.4	-0.019	NaN	NaN	NaN	NaN	N
model_pow_8	0.92	43	-1.4e+02	1.8e+02	-1.3e+02	58	-15	2.4	-0.21	0.0077	NaN	NaN	NaN	N
model_pow_9	0.87	1.7e+02	-6.1e+02	9.6e+02	-8.5e+02	4.6e+02	-1.6e+02	37	-5.2	0.42	-0.015	NaN	NaN	N
model_pow_10	0.87	1.4e+02	-4.9e+02	7.3e+02	-6e+02	2.9e+02	-87	15	-0.81	-0.14	0.026	-0.0013	NaN	N
model_pow_11	0.87	-75	5.1e+02	-1.3e+03	1.9e+03	-1.6e+03	9.1e+02	-3.5e+02	91	-16	1.8	-0.12	0.0034	N
model_pow_12	0.87	-3.4e+02	1.9e+03	-4.4e+03	5e+03	-5.2e+03	3.1e+03	-1.3e+03	3.8e+02	-80	12	-1.1	0.062	-
model_pow_13	0.86	3.2e+03	-1.8e+04	4.5e+04	-6.7e+04	6.6e+04	-4.6e+04	2.3e+04	-8.5e+03	2.2e+03	-4.5e+02	62	-5.7	0
model_pow_14	0.78	2.4e+04	-1.4e+05	3.8e+05	-6.1e+05	6.6e+05	-5e+05	2.8e+05	-1.2e+05	3.7e+04	-8.5e+03	-1.5e+03	-1.6e+02	1
model_pow_15	0.7	-3.6e+04	2.4e+05	-7.5e+05	1.4e+06	-1.7e+06	1.5e+06	-1e+06	5e+05	-1.9e+05	5.4e+04	-1.2e+04	1.9e+03	-

Weight

PENALTY

LASSO, Ridge Regression

$$argmin(y - wx)^2 + \alpha |w|$$



α

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	co
alpha_1e-15	0.96	0.22	1.1	-0.37	0.00089	0.0016	-0.00012	-6.4e-05	-6.3e-06	1.4e-06	7.8e-07	2.1e-07	4e-08	5.4
alpha_1e-10	0.96	0.22	1.1	-0.37	0.00088	0.0016	-0.00012	-6.4e-05	-6.3e-06	1.4e-06	7.8e-07	2.1e-07	4e-08	5.4
alpha_1e-08	0.96	0.22	1.1	-0.37	0.00077	0.0016	-0.00011	-6.4e-05	-6.3e-06	1.4e-06	7.8e-07	2.1e-07	4e-08	5.3
alpha_1e-05	0.96	0.5	0.6	-0.13	-0.036	-0	0	0	0	7.7e-06	1e-06	7.7e-08	0	0
alpha_0.0001	1	0.9	0.17	-0	-0.048	-0	-0	0	0	9.5e-06	5.1e-07	0	0	0
alpha_0.001	1.7	1.3	-0	-0.13	-0	-0	-0	0	0	0	0	0	1.5e-06	7.5
alpha_0.01	3.6	1.8	-0.55	-0.00056	-0	-0	-0	-0	-0	-0	-0	0	0	0
alpha_1	37	0.038	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
alpha_5	37	0.038	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
alpha_10	37	0.038	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0

HIGH SPARSITY

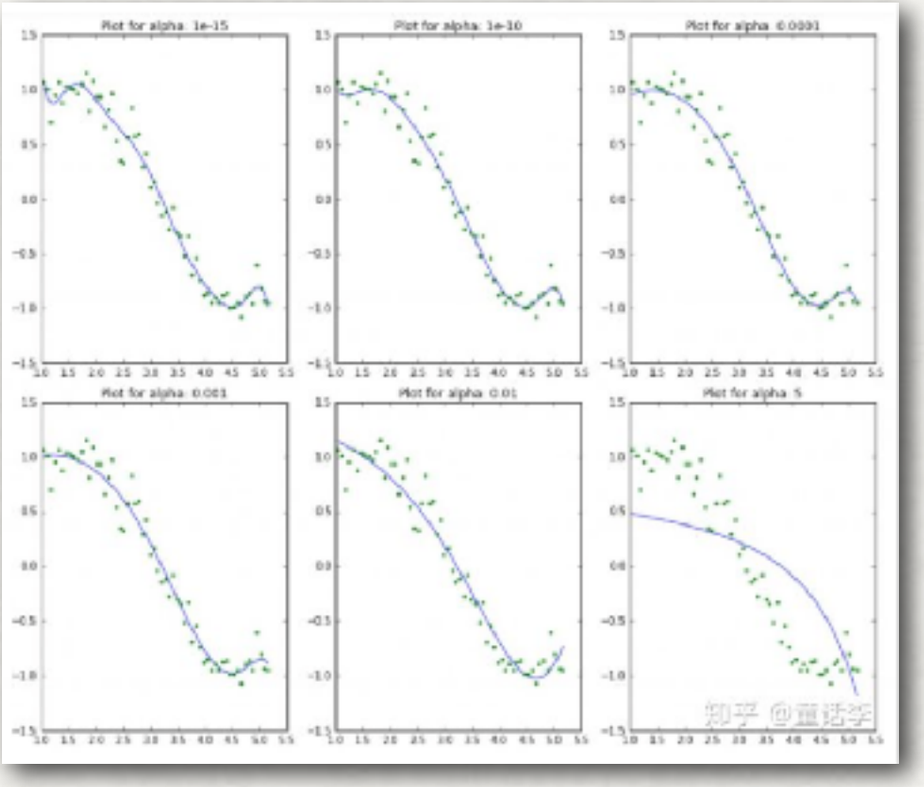
知乎 @童话学

Penalty weight

PENALTY

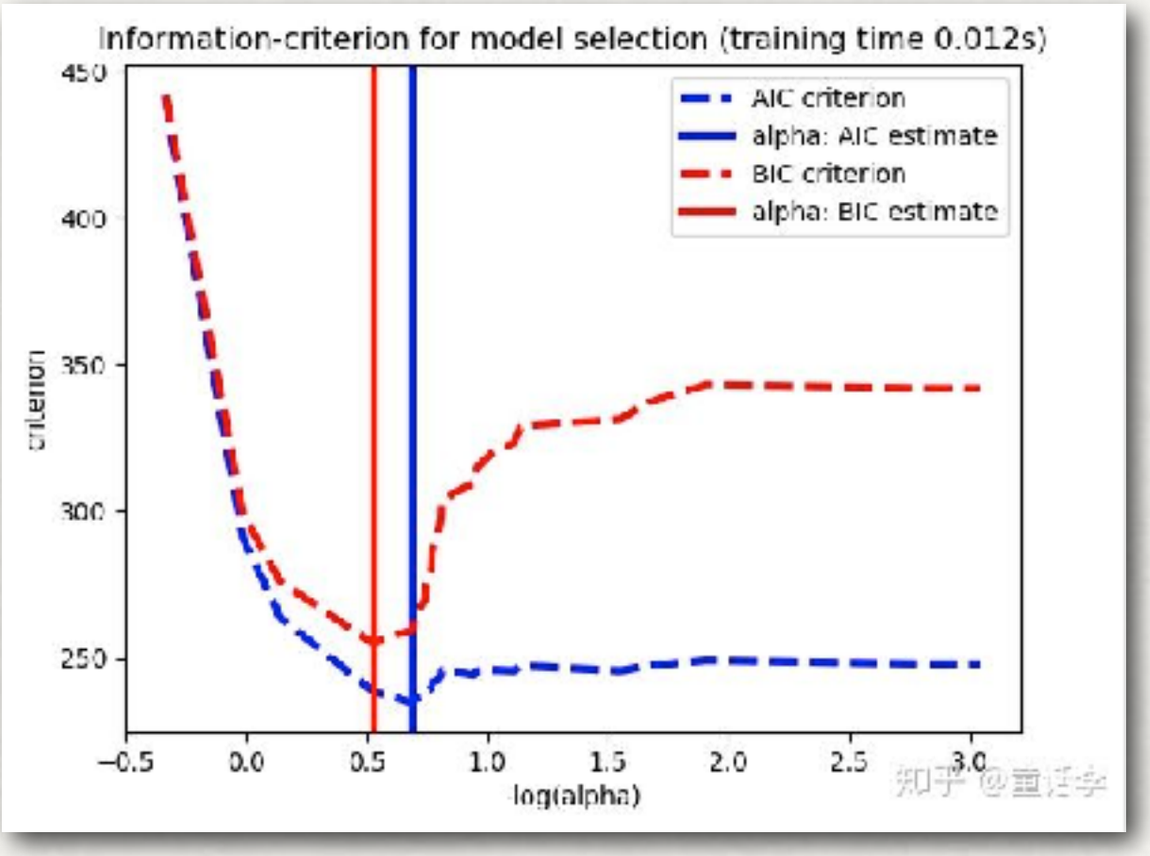
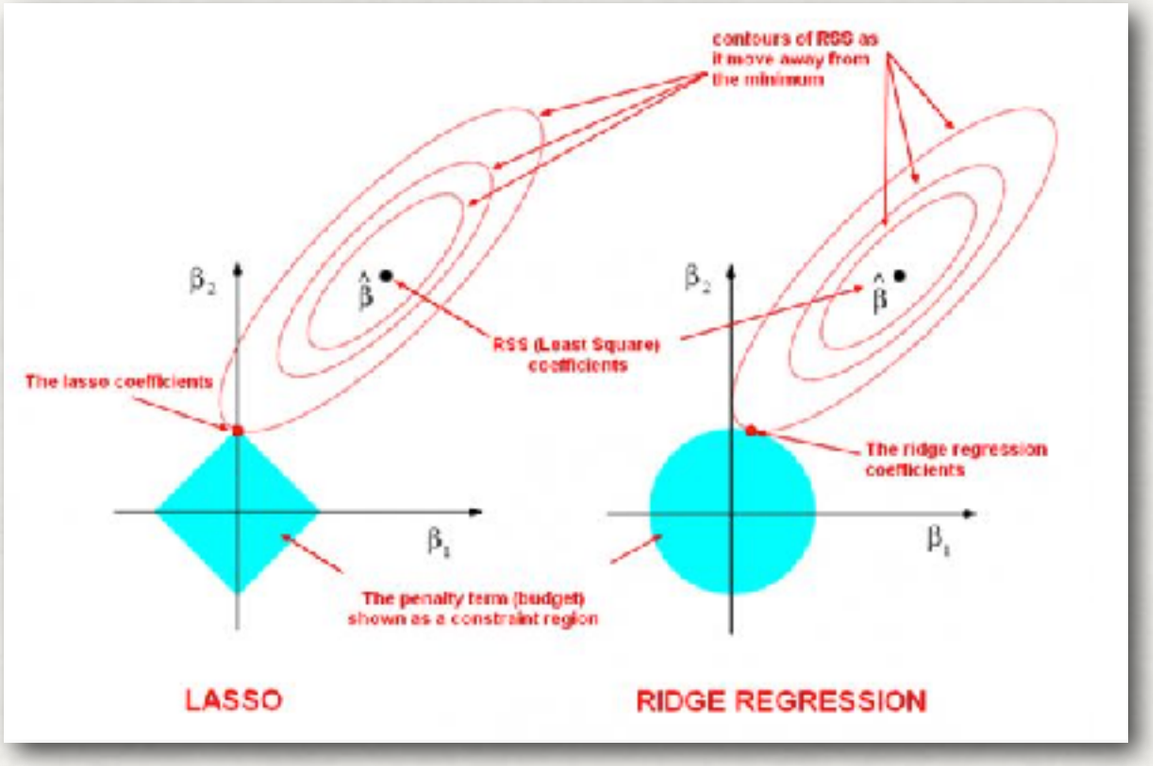
Ridge Regression

$$argmin(y - wx)^2 + \lambda w^2$$



α

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	co
alpha_1e-15	0.87	95	-3e+02	3.6e+02	-2.4e+02	66	0.96	-4.8	0.64	0.15	-0.026	-0.0054	0.00086	0.0
alpha_1e-10	0.92	11	29	31	-16	2.9	0.17	-0.091	-0.011	0.002	0.00064	2.4e-06	-2e-06	-4.
alpha_1e-08	0.95	1.3	-1.5	1.7	-0.58	0.039	0.016	0.00016	-0.00036	-5.4e-05	-2.9e-07	1.1e-06	1.9e-07	2e-
alpha_0.0001	0.96	0.56	0.55	-0.13	-0.026	-0.0028	-0.00011	4.1e-05	1.5e-05	3.7e-06	7.4e-07	1.3e-07	1.9e-08	1.9
alpha_0.001	1	0.82	0.31	-0.067	-0.02	-0.0028	-0.00022	1.8e-05	1.2e-05	3.4e-06	7.3e-07	1.3e-07	1.9e-08	1.7
alpha_0.01	1.4	1.3	0.038	-0.052	-0.01	-0.0014	-0.00013	7.2e-07	4.1e-06	1.3e-06	3e-07	5.6e-08	9e-09	1.1
alpha_1	5.6	0.97	-0.14	-0.019	-0.003	-0.00047	-7e-05	-9.9e-06	-1.3e-06	-1.4e-07	-9.3e-09	1.3e-09	7.8e-10	2.4
alpha_5	14	0.55	-0.056	-0.0085	-0.0014	-0.00024	-4.1e-05	-6.9e-06	-1.1e-06	-1.9e-07	-3.1e-08	-5.1e-09	-8.2e-10	-1.
alpha_10	18	0.4	-0.037	-0.0055	-0.00095	-0.00017	-3e-05	-5.2e-06	-9.2e-07	-1.6e-07	-2.9e-08	-3.7e-09	[illegible]	[illegible]
alpha_20	23	0.28	-0.022	-0.0034	-0.0006	-0.00011	-2e-05	-3.6e-06	-6.6e-07	-1.2e-07	-2.2e-08	-4e-09	-7.5e-10	-1.

Penalty weight

THE RIGHT ALPHA